Statistical Inference in Large Language Models

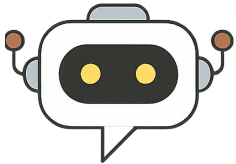
Language Watermarking: Motivation, Methods & Advances

Xiang Li

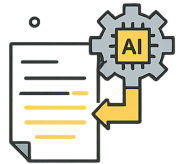
University of Pennsylvania

2025 ICSA Applied Statistics Symposium
June 15, 2025

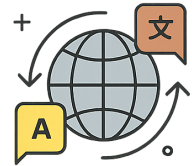
LLMs in everyday use



Chatbots

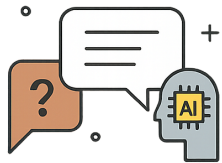


Text generration

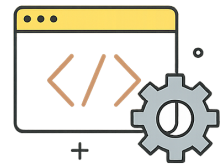


Language translation

Sentiment analysis



Question answering



Code generation

Do you trust the student?

- Did the student write this homework/paper by himself, or did an LLM lend a hand?

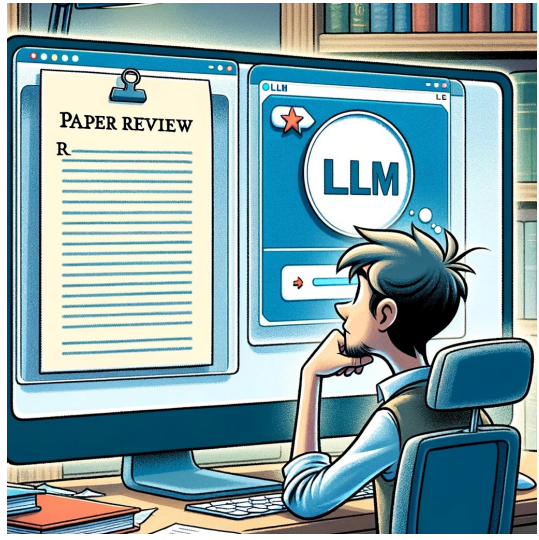
Do you trust the student?

A New Headache for Honest Students: Proving They Didn't Use A.I.

Students are resorting to extreme measures to fend off accusations of cheating, including hourslong screen recordings of their homework sessions.

Peer review or LLM-assisted review?

- Liang et al. [2024] finds that between 6.5% and 16.9% reviews of some ML conferences were substantially modified by LLMs.
- Is your paper review really your own, or did an LLM lend a hand?



Even use LLMs to write papers!

ARTICLE INFO

Keywords:
Lithium metal battery
Lithium dendrites
CuMOF-ANFs separator

ABSTRACT

Lithium metal, due to its advantages of high theoretical capacity, low density and low electrochemical reaction potential, is used as a negative electrode material for batteries and brings great potential for the next generation of energy storage systems. However, the production of lithium metal dendrites makes the battery life low and poor safety, so lithium dendrites have been the biggest problem of lithium metal batteries. This study shows that the larger specific surface area and more pore structure of Cu-based metal-organic-framework - aramid cellulose (CuMOF-ANFs) composite separator can help to inhibit the formation of lithium dendrites. After 110 cycles at 1 mA/cm^2, the discharge capacity retention rate of the Li-Cu battery using the CuMOF-ANFs separator is about 96 %. Li-Li batteries can continue to maintain low hysteresis for 2000 h at the same current density. The results show that CuMOF-ANFs composite membrane can inhibit the generation of lithium dendrites and improve the cycle stability and cycle life of the battery. The three-dimensional (3D) porous mesh structure of CuMOF-ANFs separator provides a new perspective for the practical application of lithium metal battery.

1. Introduction

Certainly, here is a possible introduction for your topic:Lithium-metal batteries are promising candidates for high-energy-density rechargeable batteries due to their low electrode potentials and high chemical stability of the separator is equally important as it ensures that the separator remains intact and does not react or degrade in the presence of the electrolyte or other battery components. A chemically stable separator helps to prevent the formation of reactive species that can further promote dendrite growth. Researchers are actively exploring

General LLM risks

Fake news

Plagiarism

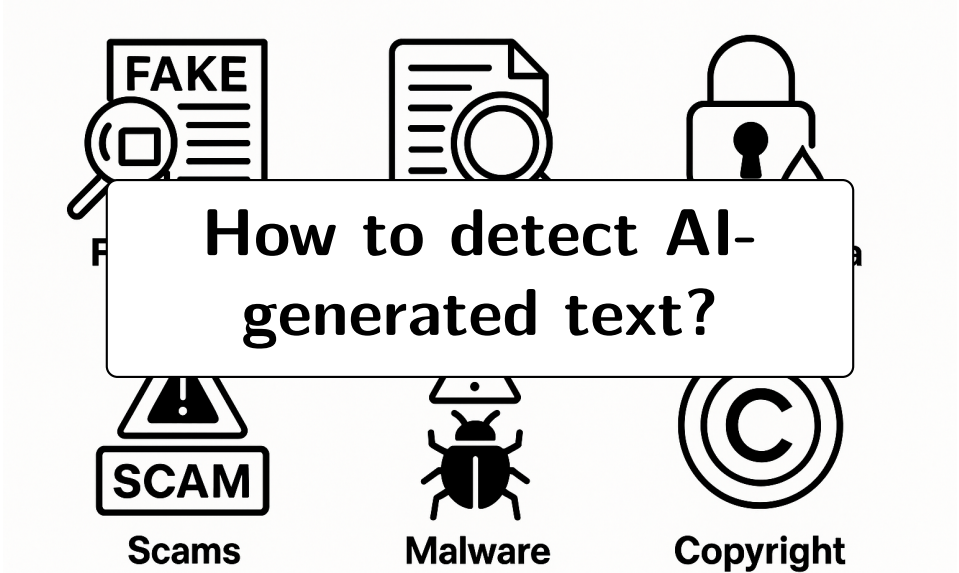
Private data leaks

Scams

Malware

Copyright

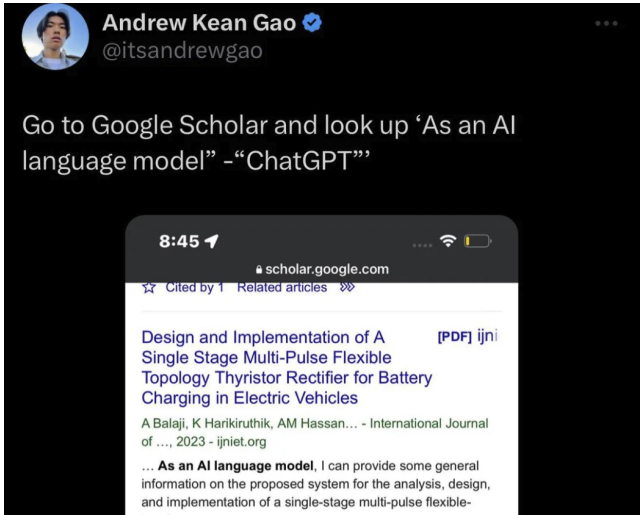
General LLM risks



How to detect AI-generated text?

Method 1

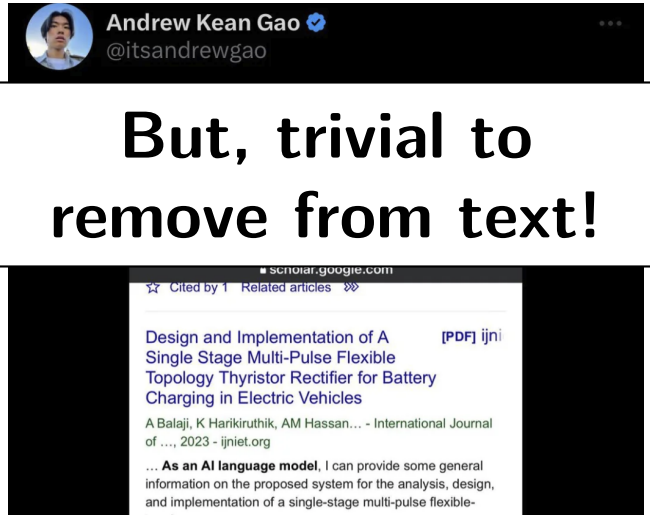
- Add prefix: “As a large language model. . . ”



How to detect AI-generated text?

Method 1

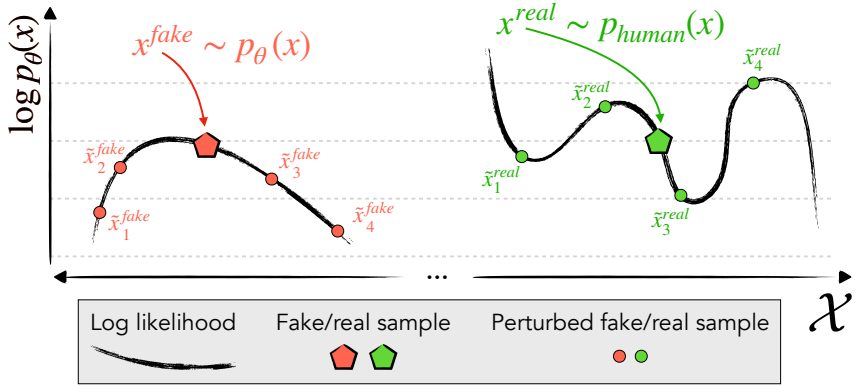
- Add prefix: "As a large language model. . . "



How to detect AI-generated text?

Method 2

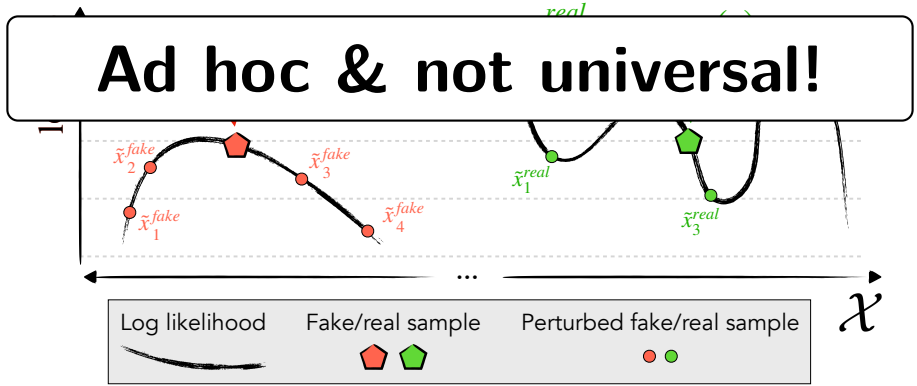
- Use linguistic pattern deviations.
- Log probability curvature (below) [Mitchell et al., 2023, Bao et al., 2023]...
- Divergent n-gram analysis [Yang et al., 2023]...



How to detect AI-generated text?

Method 2

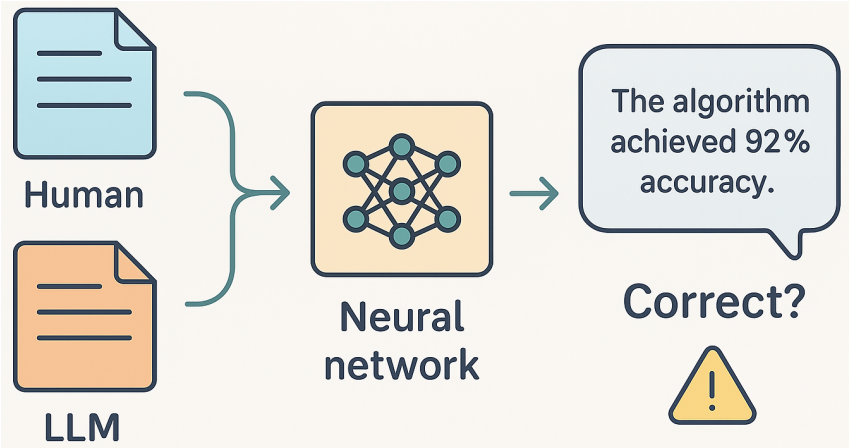
- Use linguistic pattern deviations.
- Log probability curvature (below) [Mitchell et al., 2023, Bao et al., 2023]...
- Divergent n-gram analysis [Yang et al., 2023]...



How to detect AI-generated text?

Method 3

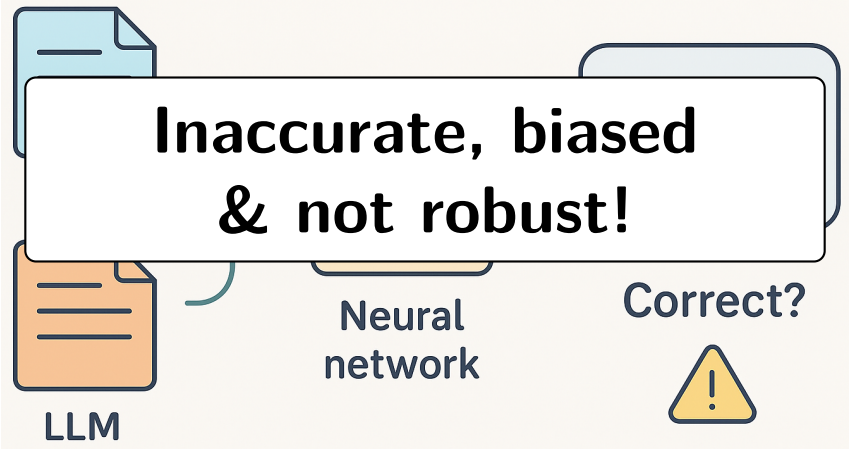
- Train classifiers [GPTZero, 2023, ZeroGPT, 2023, ...]



How to detect AI-generated text?

Method 3

- Train classifiers [GPTZero, 2023, ZeroGPT, 2023, ...]



Watermarking is a provable and practical solution!

Key insight

LLMs are probabilistic machines, and we control how they generate texts.

Watermarking is a provable and practical solution!

Key insight

LLMs are probabilistic machines, and we control how they generate texts.

A watermark embeds subtle and recoverable statistical signals into LLM-generated texts [Kirchenbauer et al., 2023].

- Creates a statistical dependency between the visible text and a hidden information.
- Unlikely to appear in human-written text.
- Applies not only to text but also to images, tables, and other data modalities.

Watermarking is a provable and practical solution!

Key insight

LLMs are probabilistic machines, and we control how they generate texts.

A waterma…erated texts [Kirc…

Efficient & provable!

- Create… …formation.
- Unlikely to appear in human-written text.
- Applies not only to text but also to images, tables, and other data modalities.

An active research area with practical importance

A Zoo of Watermarking Schemes **(since Jan 2023):**

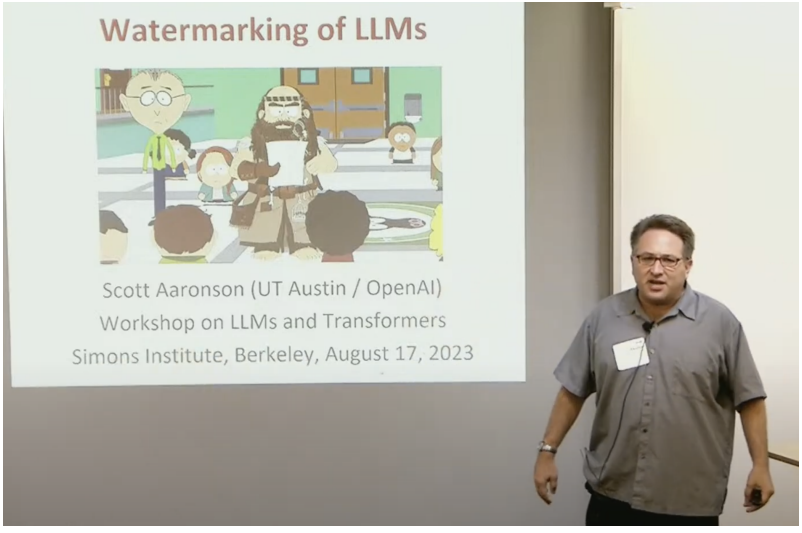
[Kirchenbauer et al., 2023, Aaronson, 2023, Kuditipudi et al., 2024, Zhao et al., 2024b, Christ et al., 2024, Wu et al., 2023, Hu et al., 2024, Kirchenbauer et al., 2024, Zhao et al., 2024a, Xie et al., 2024, Fu et al., 2024, Dathathri et al., 2024, Gloaguen et al., 2025, Abdalla and Vershynin, 2025, ...].

An active research area with practical importance

A Zoo of Watermarking Schemes **(since Jan 2023):**
[Kirchenbauer et al., 2023, Aaronson, 2023, Kuditipudi et al., 2024, Zhao et al., 2024b, Christ et al., 2024, Wu et al., 2023, Hu et al., 2024, Kirchenbauer et al., 2024, Zhao et al., 2024a, Xie et al., 2024, Fu et al., 2024, Dathathri et al., 2024, Gloaguen et al., 2025, Abdalla and Vershynin, 2025, ...].

- **Open-source toolkits** have been developed to support research [Pan et al., 2024].
- **Large-scale empirical studies** benchmark watermarking methods [Fernandez et al., 2023a, Gloaguen et al., 2025, Piet et al., 2023, ...].
- **Industry commitment**: OpenAI, Google, Meta, and others pledge to watermark AI-generated content.

OpenAI's watermark: Gumbel-max [Aaronson, 2023]



https://simons.berkeley.edu/talks/scott-aaronson-ut-austin-openai-2023-08-17

Google's watermark: SynthID [Dathathri et al., 2024]

Article | Open access | Published: 23 October 2024

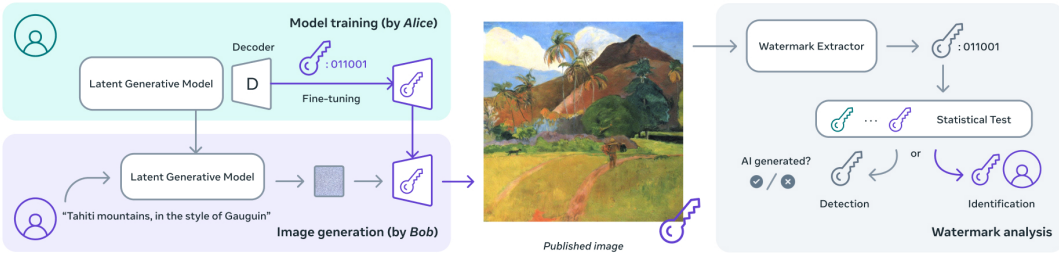
Scalable watermarking for identifying large language model outputs

Sumanth Dathathri ✉, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Al Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, Taylan Cemgil, Zahra Ahmed, Kitty Stacpoole, Ilia Shumailov, Ciprian Baetu, Sven Gowal, Demis Hassabis & Pushmeet Kohli ✉

Nature **634**, 818–823 (2024) | Cite this article

https://deepmind.google/science/synthid/

Meta's watermark: Stable Signature [Fernandez et al., 2023b]

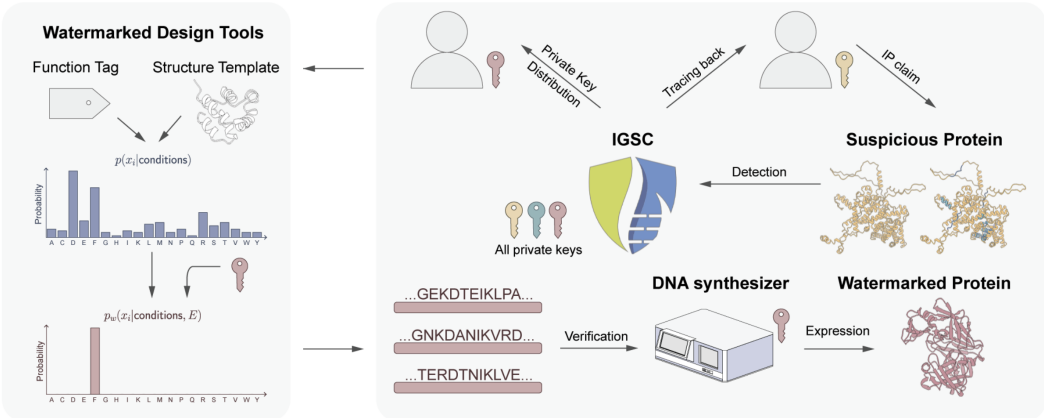


Note that this is for images.

https://ai.meta.com/blog/stable-signature-watermarking-generative-ai/

Watermark for biosecurity: Protein design

- Analogously, Chen et al. [2025] generate amino acids autoregressively and watermark them like text.



Python package: https://github.com/poseidonchan/ProteinWatermark

A field where statistics can make meaningful impact

Statistical perspectives

- **Type I error** – falsely flagging human prose as AI-generated
- **Type II error** – failing to detect AI-generated text
- **Prevalence** – estimating the share of AI-generated texts
- **Localization** – identifying which subtexts are AI-generated
- ...

Recent advances

- Optimal detection rules
- Robust detection under mixture model
- Goodness-of-fit tests
- Proportion estimation
- Token-level localization
- ...

In this short course

Why consider watermarks

How to embed a watermark

How to detect the watermark

Recent statistical advances in watermarking

Concluding remarks

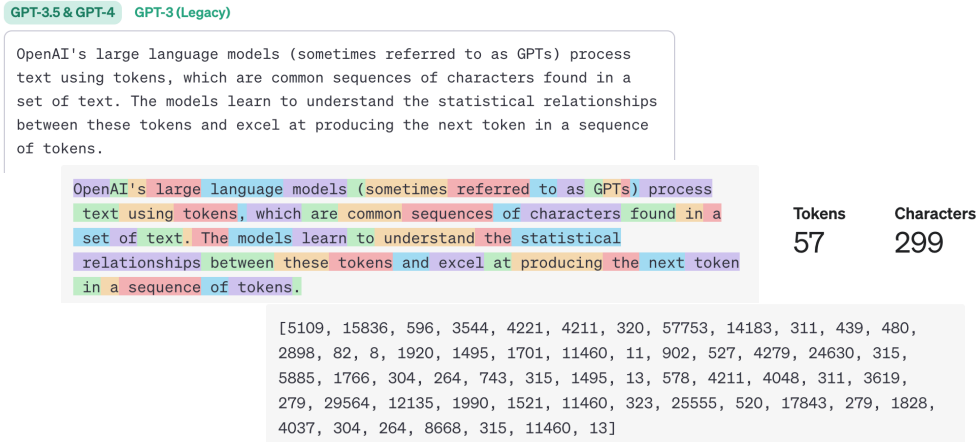
Outlines

Tokens: Smallest units of LLM generation

- LLMs generate text by gathering many small units, called “tokens”.
- Tokens can be words, parts of words, or even punctuation marks.

Tokens: Smallest units of LLM generation

- LLMs generate text by gathering many small units, called “tokens”.
- Tokens can be words, parts of words, or even punctuation marks.



https://platform.openai.com/tokenizer

Autoregresive generation: How LLMs combine tokens

Denote the vocabulary by $\mathcal{W} = \{1, \ldots, K\}$, a token therein by w_t, and a text by $w_{<t} := w_1 \cdots w_{t-1}$.

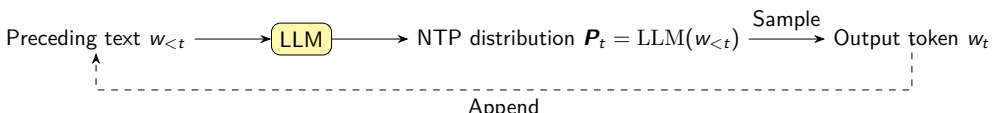
- **Large vocabulary**: $\mathcal{W}$ is large in practice; $K = 50,257$ for GPT-2/3.5, $= 32,000$ for LLaMa models, and $= 128,000$ for DeepSeek-R1.

Autoregresive generation: How LLMs combine tokens

Denote the vocabulary by $\mathcal{W} = \{1, \ldots, K\}$, a token therein by w_t, and a text by $w_{<t} := w_1 \cdots w_{t-1}$.

- **Large vocabulary**: $\mathcal{W}$ is large in practice; $K = 50,257$ for GPT-2/3.5, $= 32,000$ for LLaMa models, and $= 128,000$ for DeepSeek-R1.
- **Autoregresiveness**: An LLM generates each token sequentially by sampling from a probability distribution conditioned on previous tokens:

$$w_t \sim \boldsymbol{P}_t \quad \text{where} \quad \boldsymbol{P}_t = \mathrm{LLM}(w_{<t}) \quad \text{is a distribution on} \quad \mathcal{W}.$$

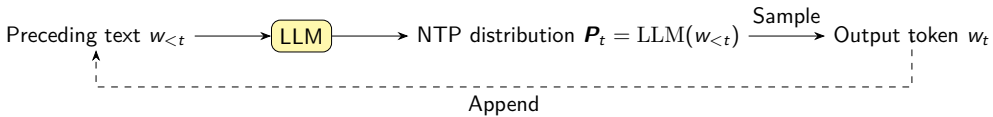


Autoregresive generation: How LLMs combine tokens

Denote the vocabulary by $\mathcal{W} = \{1, \ldots, K\}$, a token therein by w_t, and a text by $w_{<t} := w_1 \cdots w_{t-1}$.

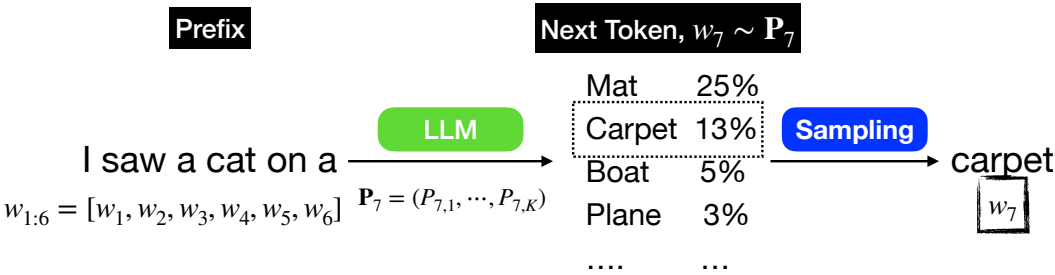
- **Large vocabulary**: $\mathcal{W}$ is large in practice; $K = 50,257$ for GPT-2/3.5, $= 32,000$ for LLaMa models, and $= 128,000$ for DeepSeek-R1.
- **Autoregresiveness**: An LLM generates each token sequentially by sampling from a probability distribution conditioned on previous tokens:

$$w_t \sim \boldsymbol{P}_t \quad \text{where} \quad \boldsymbol{P}_t = \mathrm{LLM}(w_{<t}) \quad \text{is a distribution on} \quad \mathcal{W}.$$

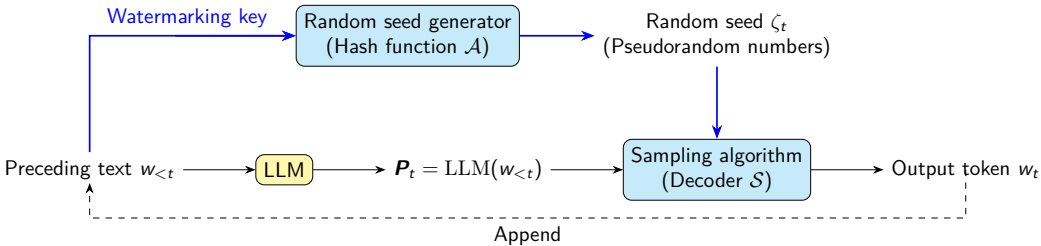


- **Limited access**: The distribution $\boldsymbol{P}_t$ is referred to next-token prediction (NTP) distribution, which is unknown since it depends on unknown system/user prompts.

Autoregresive generation: Example

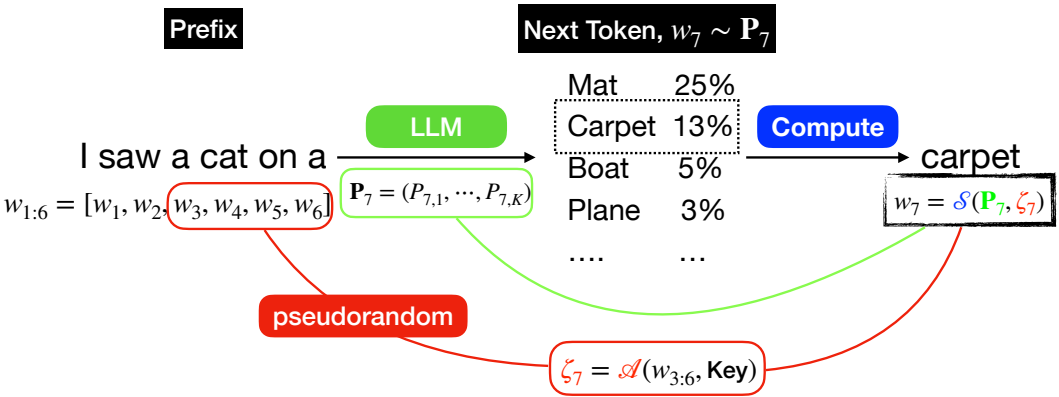


Watermarked generation: Procedure



- Mathematical speaking: $\zeta_t = \mathcal{A}(w_{<t}, \mathrm{Key})$ and $w_t = \mathcal{S}(\boldsymbol{P}_t, \zeta_t)$.
- A watermark is defined by $(\mathcal{A}, \mathcal{S}, \mathrm{Key})$.
- Watermark signal is the dependence of each w_t on ζ_t.

Watermarked generation: Example



Comments on pseudorandom numbers

$\mathcal{A}$(public data, private info) computes the pseudorandom number:

- Public data = prior tokens $w_{<t}$ or a segment $w_{(t-m):t}$ [Kirchenbauer et al., 2023].
- Private info = secret key, denoted Key.

Comments on pseudorandom numbers

$\mathcal{A}(\text{public data}, \text{private info})$ computes the pseudorandom number:

- Public data = prior tokens $w_{<t}$ or a segment $w_{(t-m):t}$ [Kirchenbauer et al., 2023].
- Private info = secret key, denoted Key.

Property (*Soundness of pseudorandomness*)

1. $\zeta_t = \mathcal{A}(w_{1:t-1}, \mathrm{Key})$ for $t = 1, \ldots, n$ are iid copies of a known random variable ζ
2. ζ_t is statistically independent of $w_{1:t-1}$
3. Computationally infeasible to infer Key from ζ_t and $w_{<t}$

Comments on pseudorandom numbers

$\mathcal{A}(\text{public data}, \text{private info})$ computes the pseudorandom number:

- Public data = prior tokens $w_{<t}$ or a segment $w_{(t-m):t}$ [Kirchenbauer et al., 2023].
- Private info = secret key, denoted Key.

Property (*Soundness of pseudorandomness*)

1. $\zeta_t = \mathcal{A}(w_{1:t-1}, \mathrm{Key})$ for $t = 1, \ldots, n$ are iid copies of a known random variable ζ
2. ζ_t is statistically independent of $w_{1:t-1}$
3. Computationally infeasible to infer Key from ζ_t and $w_{<t}$

- Well-developed in theoretical computer science [Barak, 2021].
  - Pseudorandom numbers are "reproducible" ***iff*** Key is known.
  - Key will be shared with the verifier through a secure protocol.

Comments on pseudorandom numbers

$\mathcal{A}(\text{public data}, \text{private info})$ computes the pseudorandom number:

- Public data = prior tokens $w_{<t}$ or a segment $w_{(t-m):t}$ [Kirchenbauer et al., 2023].
- Private info = secret key, denoted Key.

Property (*Soundness of pseudorandomness*)

1. $\zeta_t = \mathcal{A}(w_{1:t-1}, \mathrm{Key})$ for $t = 1, \ldots, n$ are iid copies of a known random variable ζ
2. ζ_t is statistically independent of $w_{1:t-1}$
3. Computationally infeasible to infer Key from ζ_t and $w_{<t}$

- Well-developed in theoretical computer science [Barak, 2021].
  - Pseudorandom numbers are "reproducible" ***iff*** Key is known.
  - Key will be shared with the verifier through a secure protocol.
- Theory assumes true randomness; Implementation use deterministic generation.
- Similar to setting a *seed* for reproducibility in simulations.

Pseudocode for watermark embedding

Algorithm Watermarked LLM Generation

Inputs: a watermark $(\mathcal{S}, \mathcal{A}, \mathrm{Key})$ and a language model.
Load the language model $\mathrm{LLM}(\cdot)$.
Receive the user prompt s and feed it to the model to generate a continuation.
Initialize $w_{<1} = s$ and set n the maximum length.
for step $t = 1, 2, \cdots n$ **do**
    Compute the NTP distribution: $\boldsymbol{P}_t = \mathrm{LLM}(w_{<t})$.
    Compute the pseudorandom number: $\zeta_t = \mathcal{A}(w_{<t}, \mathrm{Key})$.
    Compute the next token: $w_t = \mathcal{S}(\boldsymbol{P}_t, \zeta_t)$.
    Append the history: $w_{<(t+1)} = w_{<t} w_t$. ▷ Autoregressive
return $w_{\leq n}$.

A high-level intro of watermark detection

$$H_0 : w_{\leq n} \text{ is human written} \quad \text{v.s.} \quad H_0 : w_{\leq n} \text{ is LLM-generated.}$$

- **Human-written text**: w_t is independent of ζ_t as humans don't know $\mathcal{A}$ and Key.
- **LLM-generated text**: w_t depends on ζ_t via the decoder function $\mathcal{S}$.

A high-level intro of watermark detection

$$H_0 : w_{\leq n} \text{ is human written} \quad \text{v.s.} \quad H_0 : w_{\leq n} \text{ is LLM-generated.}$$

- **Human-written text**: w_t is independent of ζ_t as humans don't know $\mathcal{A}$ and Key.
- **LLM-generated text**: w_t depends on ζ_t via the decoder function $\mathcal{S}$.

Take-away

Watermarking couples each token w_t and a psedorandom ζ_t, altering their joint distribution.

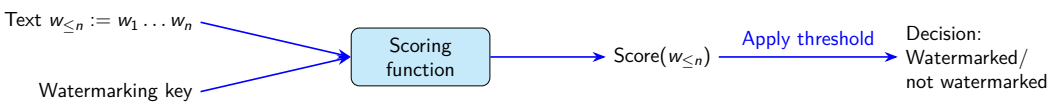
A high-level intro of watermark detection

$$H_0 : w_{\leq n} \text{ is human written} \quad \text{v.s.} \quad H_0 : w_{\leq n} \text{ is LLM-generated.}$$

- **Human-written text**: w_t is independent of ζ_t as humans don't know $\mathcal{A}$ and Key.
- **LLM-generated text**: w_t depends on ζ_t via the decoder function $\mathcal{S}$.

Take-away

Watermarking couples each token w_t and a psedorandom ζ_t, altering their joint distribution.



A self-contained watermark python demo

- **Download**:
  https://github.com/lx10077/WatermarkFramework/blob/main/watermark_demo.py
- **How to use**:
  `python watermark_demo.py -temp 1 -alpha 0.01 -model facebook/opt-1.3b`

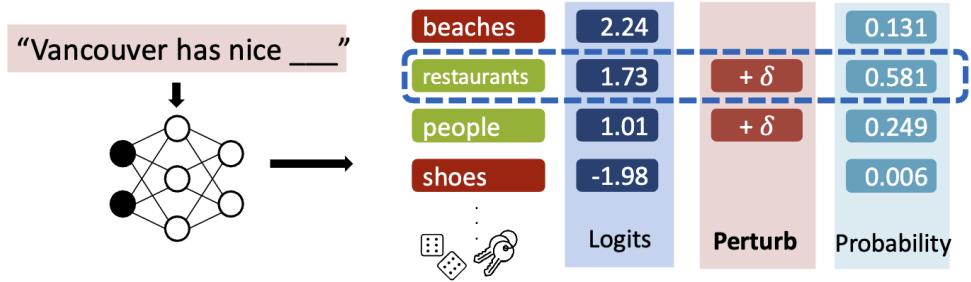
```
tokenizer = AutoTokenizer.from_pretrained(args.model) # Load tokenizer which convers text to a sequence of
# Ensure the tokenizer has a pad token
if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token
model = AutoModelForCausalLM.from_pretrained(
    args.model,
    device_map="auto",             # Automatically place model layers on available GPU(s)
    torch_dtype=torch.float16      # (Optional) Set tensor data type to float16 for faster computation
)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu") # Use GPU if possible
print(f"Using device: {device}")
model = model.to(device) # Move the model to GPU, otherwise it is default on CPU
model.eval()

# Load the first 200 samples from the AG News dataset.
# You could also test your own questions or queries. The model will continue to write after your given tex
# To that end, simply change the following 'raw_texts' with a list of your questions.
dataset = load_dataset("ag_news", split="train[:200]")
```

Some examples of watermarks

Green-red list watermark [Kirchenbauer et al., 2023]

- Randomly split vocabulary in to green (favored) and red (disfavored) parts.
- Secretly boost the prob. of green tokens, i.e., $P^{\text{wm}}_{\text{green}} \propto e^{\delta} P_{\text{green}}$ and $P^{\text{wm}}_{\text{red}} \propto P_{\text{red}}$.
- If the observed frequency of green tokens is larger than expected, claim watermarked.

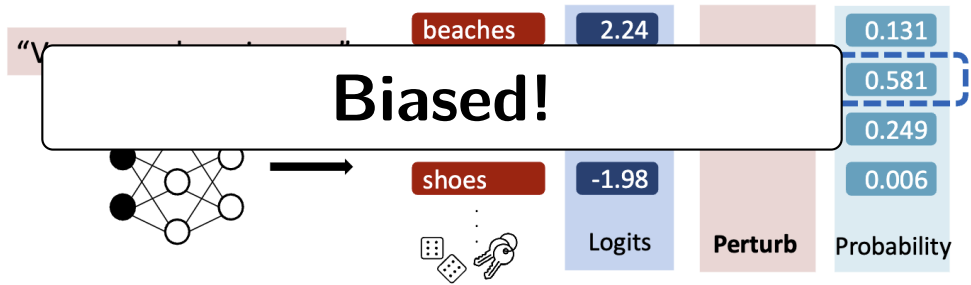


Kirchenbauer et al. (2023) A Watermark for Large Language Models
Zhao et al. (2023) Provable Robust Watermarking for AI-Generated Text

Figure from the tutorial: https://leililab.github.io/llm_watermark_tutorial/

Green-red list watermark [Kirchenbauer et al., 2023]

- Randomly split vocabulary in to green (favored) and red (disfavored) parts.
- Secretly boost the prob. of green tokens, i.e., $P^{\mathrm{wm}}_{\mathrm{green}} \propto e^{\delta} P_{\mathrm{green}}$ and $P^{\mathrm{wm}}_{\mathrm{red}} \propto P_{\mathrm{red}}$.
- If the observed frequency of green tokens is larger than expected, claim watermarked.



Kirchenbauer et al. (2023) A Watermark for Large Language Models
Zhao et al. (2023) Provable Robust Watermarking for AI-Generated Text

Figure from the tutorial: https://leililab.github.io/llm_watermark_tutorial/

The simplest unbiased watermark: Baby watermark

Definition (*Unbiasedness*)

A watermark is unbiased if the marginal distribution of w in (w, ζ) is still $\boldsymbol{P}$, i.e.,

$$\mathbb{P}_\zeta(\mathcal{S}(\boldsymbol{P}, \zeta) = w) = P_w \text{ for any } \boldsymbol{P} \text{ and } w \in \mathcal{W}.$$

The simplest unbiased watermark: Baby watermark

Definition (*Unbiasedness*)

A watermark is unbiased if the marginal distribution of w in (w, ζ) is still $\boldsymbol{P}$, i.e.,

$$\mathbb{P}_\zeta(\mathcal{S}(\boldsymbol{P}, \zeta) = w) = P_w \text{ for any } \boldsymbol{P} \text{ and } w \in \mathcal{W}.$$

- Let $\mathcal{W} = \{0, 1\}$, $\boldsymbol{P}_t = (P_{t,0}, P_{t,1})$, and ζ_t be i.i.d. copies of $\mathcal{U}(0, 1)$.
- Decoder

$$w_t = \mathcal{S}(\boldsymbol{P}_t, \zeta_t) = \begin{cases} 0 & \text{if } \zeta_t \leq P_{t,0} \\ 1 & \text{otherwise.} \end{cases}$$

The simplest unbiased watermark: Baby watermark

Definition (*Unbiasedness*)

A watermark is unbiased if the marginal distribution of w in (w, ζ) is still $\boldsymbol{P}$, i.e.,

$$\mathbb{P}_\zeta(\mathcal{S}(\boldsymbol{P}, \zeta) = w) = P_w \text{ for any } \boldsymbol{P} \text{ and } w \in \mathcal{W}.$$

- Let $\mathcal{W} = \{0, 1\}$, $\boldsymbol{P}_t = (P_{t,0}, P_{t,1})$, and ζ_t be i.i.d. copies of $\mathcal{U}(0, 1)$.
- Decoder

$$w_t = \mathcal{S}(\boldsymbol{P}_t, \zeta_t) = \begin{cases} 0 & \text{if } \zeta_t \leq P_{t,0} \\ 1 & \text{otherwise.} \end{cases}$$

- If ζ_t is large, w_t is more likely to be 1.
- Statistic score for detection: $\sum_{t=1}^{n}(2w_t - 1)(2\zeta_t - 1)$.

The simplest unbiased watermark: Baby watermark

Definition (*Unbiasedness*)

A watermark is unbiased if the marginal distribution of w in (w, ζ) is still $\boldsymbol{P}$, i.e.,

$$\mathbb{P}_\zeta(\mathcal{S}(\boldsymbol{P}, \zeta) = w) = P_w \text{ for any } \boldsymbol{P} \text{ and } w \in \mathcal{W}.$$

- Let $\mathcal{W} = \{0, 1\}$, $\boldsymbol{P}_t = (P_{t,0}, P_{t,1})$, and ζ_t be i.i.d. copies of $\mathcal{U}(0, 1)$.
- Decoder

$$w_t = \mathcal{S}(\boldsymbol{P}_t, \zeta_t) = \begin{cases} 0 & \text{if } \zeta_t \leq P_{t,0} \\ 1 & \text{otherwise.} \end{cases}$$

- If ζ_t is large, w_t is more likely to be 1.
- Statistic score for detection: $\sum_{t=1}^{n}(2w_t - 1)(2\zeta_t - 1)$.

Take-away

Statistically, an unbiased watermark is basically a sampling method from each $\boldsymbol{P}_t$.

First unbiased watermark: Gumbel-max [Aaronson, 2023]

Definition (*Gumbel-max trick [Gumbel, 1948]*)

Let $U_1, \ldots, U_K$ be i.i.d. copies of $\mathcal{U}(0,1)$. Then,

$$\arg\max_{w \in \mathcal{W}} \frac{\log U_w}{P_w} \sim \boldsymbol{P} \equiv (P_w)_{w \in \mathcal{W}}.$$

First unbiased watermark: Gumbel-max [Aaronson, 2023]

Definition (*Gumbel-max trick [Gumbel, 1948]*)

Let $U_1, \dots, U_K$ be i.i.d. copies of $\mathcal{U}(0,1)$. Then,

$$\arg\max_{w \in \mathcal{W}} \frac{\log U_w}{P_w} \sim \boldsymbol{P} \equiv (P_w)_{w \in \mathcal{W}}.$$

Proof.

The decoder rule is equivalent to choosing the w that minimizes $\frac{1}{P_w} \ln \frac{1}{U_w}$. If U_w is uniform, $\frac{1}{P_w} \ln \frac{1}{U_w}$ is $\mathrm{Exp}(P_w)$, an exponential random variable with rate P_w and mean $1/P_w$. The minimum of $\mathrm{Exp}(P)$ and $\mathrm{Exp}(Q)$, is again $\mathrm{Exp}(P+Q)$. Thus we can reduce to the case of $K = 2$ tokens, for which the result can be verified by doing the integral, i.e.,

$$\mathbb{P}(\mathrm{Exp}(P) \leq \mathrm{Exp}(Q)) = \frac{P}{P+Q}.$$

□

First unbiased watermark: Gumbel-max [Aaronson, 2023]

Definition (*Gumbel-max trick [Gumbel, 1948]*)

Let $U_1, \ldots, U_K$ be i.i.d. copies of $\mathcal{U}(0,1)$. Then,

$$\arg\max_{w \in \mathcal{W}} \frac{\log U_w}{P_w} \sim \boldsymbol{P} \equiv (P_w)_{w \in \mathcal{W}}.$$

Definition (*Gumbel-max watermark*)

With $\zeta_t = (U_t, \ldots, U_K) = \mathcal{A}(w_{<t}, \mathrm{Key})$ (often depending on the last 5 tokens),

$$w_t = \mathcal{S}^{\mathrm{gum}}(\boldsymbol{P}_t, \zeta_t) = \arg\max_{w \in \mathcal{W}} \frac{\log U_{t,w}}{P_{t,w}}.$$

- Embedded signal: selected U_{w_t} tends to be larger.
- Implemented internally at OpenAI.

Many other unbiased watermarks

- Binary undetectable watermark [Christ et al., 2024].
- Inverse transform watermark [Kuditipudi et al., 2024].
- Variants of green-red lists: [Hu et al., 2024, Xie et al., 2024].
- WaterMax [Giboulot and Teddy, 2024].
- SynthID by tournament sampling [Dathathri et al., 2024].
- ...

There are also many biased watermarks, which are beyond this short course. See surveys in [Ji et al., 2025, Zhao et al., 2025].

Outlines

Hypothesis testing formulation

Human-written

When text is written by a human, w_t, ζ_t are independent, since the human simply cannot compute ζ_t.

LLM-generated

When text is generated by the LLM, w_t, ζ_t are dependent because of the decoder: $w_t = \mathcal{S}(\boldsymbol{P}_t, \zeta_t)$.

Hypothesis testing formulation

Human-written

When text is written by a human, w_t, ζ_t are independent, since the human simply cannot compute ζ_t.

LLM-generated

When text is generated by the LLM, w_t, ζ_t are dependent because of the decoder: $w_t = \mathcal{S}(\boldsymbol{P}_t, \zeta_t)$.

1. Can always compute $\zeta_t = \mathcal{A}(w_{<t}, \mathrm{Key})$, which are iid copies of ζ.
2. Dataset $=$ tokens $w_{\leq n} := w_1 w_2 \cdots w_n$ $+$ pseudorandom $\zeta_{\leq n} := \zeta_1 \zeta_2 \cdots \zeta_n$.
3. All the NTP distributions $\boldsymbol{P}_{\leq n} := \boldsymbol{P}_1 \boldsymbol{P}_2 \cdots \boldsymbol{P}_n$ are unknown.

Hypothesis testing formulation

Human-written

When text is written by a human, w_t, ζ_t are independent, since the human simply cannot compute ζ_t.

LLM-generated

When text is generated by the LLM, w_t, ζ_t are dependent because of the decoder: $w_t = \mathcal{S}(\boldsymbol{P}_t, \zeta_t)$.

1. Can always compute $\zeta_t = \mathcal{A}(w_{<t}, \mathrm{Key})$, which are iid copies of ζ.
2. Dataset = tokens $w_{\leq n} := w_1 w_2 \cdots w_n$ + pseudorandom $\zeta_{\leq n} := \zeta_1 \zeta_2 \cdots \zeta_n$.
3. All the NTP distributions $\boldsymbol{P}_{\leq n} := \boldsymbol{P}_1 \boldsymbol{P}_2 \cdots \boldsymbol{P}_n$ are unknown.

$H_0 : w_{1:n}$ by human

$(w_t, \zeta_t) \mid (w_{<t}, \zeta_{<t}) \overset{d}{=} \boldsymbol{P}_t \times \zeta$

$H_1 : w_{1:n}$ by watermarked LLM

$(w_t, \zeta_t) \mid (w_{<t}, \zeta_{<t}) \overset{d}{=} (\mathcal{S}(\zeta, \boldsymbol{P}_t), \zeta)$

A challenge: Unknown NTP distributions

$H_0 : w_{1:n}$ is by human *vs* $H_1 : w_{1:n}$ is by watermarked LLM

Hypothesis testing

- Under H_0, $(w_t, \zeta_t) \mid (w_{1:t-1}, \zeta_{1:t-1}) \overset{d}{=} \boldsymbol{P}_t \times \zeta$
- Under H_1, $(w_t, \zeta_t) \mid (w_{1:t-1}, \zeta_{1:t-1}) \overset{d}{=} (\mathcal{S}(\zeta, \boldsymbol{P}_t), \zeta)$

Neyman-Pearson lemma resorts to the log-likelihood ratio test:

$$\frac{\mathbb{P}_{H_1}(w_{1:n}, \zeta_{1:n})}{\mathbb{P}_{H_0}(w_{1:n}, \zeta_{1:n})} = \begin{cases} \frac{1}{P_{1,w_1} \cdots P_{n,w_n}} & \text{if } \mathcal{S}(\boldsymbol{P}_t, \zeta_t) = w_t \text{ for all } t \\ 0 & \text{otherwise} \end{cases}$$

A challenge: Unknown NTP distributions

$H_0 : w_{1:n}$ is by human *vs* $H_1 : w_{1:n}$ is by watermarked LLM

Hypothesis testing

- Under H_0, $(w_t, \zeta_t) \mid (w_{1:t-1}, \zeta_{1:t-1}) \overset{d}{=} \boldsymbol{P}_t \times \zeta$
- Under H_1, $(w_t, \zeta_t) \mid (w_{1:t-1}, \zeta_{1:t-1}) \overset{d}{=} (\mathcal{S}(\zeta, \boldsymbol{P}_t), \zeta)$

Neyman-Pearson lemma resorts to the log-likelihood ratio test:

$$\frac{\mathbb{P}_{H_1}(w_{1:n}, \zeta_{1:n})}{\mathbb{P}_{H_0}(w_{1:n}, \zeta_{1:n})} = \begin{cases} \frac{1}{P_{1,w_1} \cdots P_{n,w_n}} & \text{if } \mathcal{S}(\boldsymbol{P}_t, \zeta_t) = w_t \text{ for all } t \\ 0 & \text{otherwise} \end{cases}$$

- But $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$ as nuisance are unknown to the verifier, and worse, are varying!

A practical approach: Pivot under the null [Li et al., 2025a]

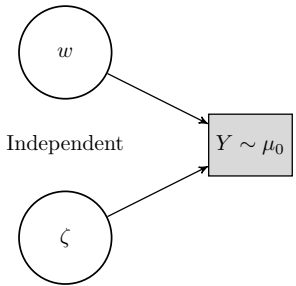
Find a pivotal statistic $Y_t = Y(w_t, \zeta_t)$ such that

- Under H_0, $Y_t \sim \mu_0$, regardless of $\boldsymbol{P}_t$
- Under H_1, $Y_t \sim Y(\mathcal{S}(\zeta_t, \boldsymbol{P}_t), \zeta_t)$, whose distribution is denoted $\mu_{1,\boldsymbol{P}_t}$

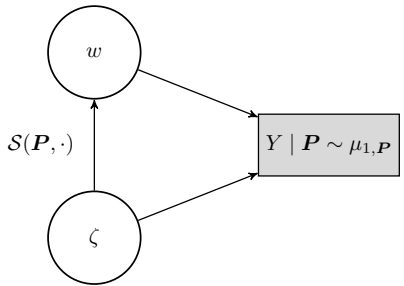
A practical approach: Pivot under the null [Li et al., 2025a]

Find a pivotal statistic $Y_t = Y(w_t, \zeta_t)$ such that

- Under H_0, $Y_t \sim \mu_0$, regardless of $\boldsymbol{P}_t$
- Under H_1, $Y_t \sim Y(\mathcal{S}(\zeta_t, \boldsymbol{P}_t), \zeta_t)$, whose distribution is denoted $\mu_{1,\boldsymbol{P}_t}$



H_0

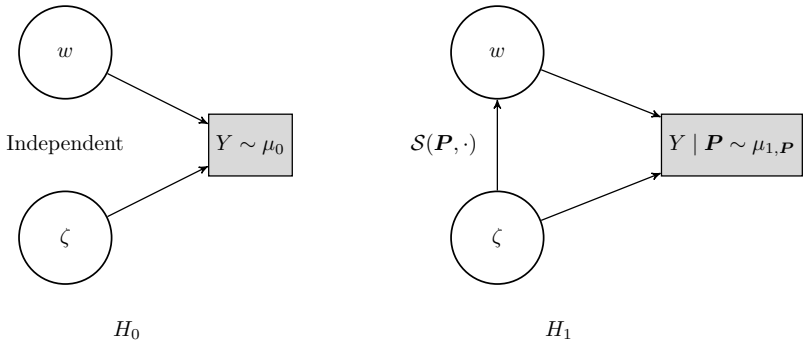


H_1

A practical approach: Pivot under the null [Li et al., 2025a]

Find a pivotal statistic $Y_t = Y(w_t, \zeta_t)$ such that

- Under H_0, $Y_t \sim \mu_0$, regardless of $\boldsymbol{P}_t$
- Under H_1, $Y_t \sim Y(\mathcal{S}(\zeta_t, \boldsymbol{P}_t), \zeta_t)$, whose distribution is denoted $\mu_{1,\boldsymbol{P}_t}$



Example: $Y_t = (2w_t - 1)(2\zeta_t - 1) \sim \mathcal{U}(-1, 1)$ under H_0 for the baby watermark.

Hypothesis testing via pivoting

Find a pivotal statistic $Y_t = Y(w_t, \zeta_t)$ such that

- Under H_0, $Y_t \sim \mu_0$, regardless of $\boldsymbol{P}_t$
- Under H_1, $Y_t \sim Y(\mathcal{S}(\zeta_t, \boldsymbol{P}_t), \zeta_t)$, whose distribution is denoted $\mu_{1,\boldsymbol{P}_t}$

Hypothesis testing via pivoting

$$H_0 : Y_t \overset{iid}{\sim} \mu_0, \ t = 1, \ldots, n \qquad \text{vs} \qquad H_1 : Y_t | \boldsymbol{P}_t \sim \mu_{1,P_t}, \ t = 1, \ldots, n$$

Hypothesis testing via pivoting

Find a pivotal statistic $Y_t = Y(w_t, \zeta_t)$ such that

- Under H_0, $Y_t \sim \mu_0$, regardless of $\boldsymbol{P}_t$
- Under H_1, $Y_t \sim Y(\mathcal{S}(\zeta_t, \boldsymbol{P}_t), \zeta_t)$, whose distribution is denoted $\mu_{1,\boldsymbol{P}_t}$

Hypothesis testing via pivoting

$$H_0 : Y_t \overset{iid}{\sim} \mu_0,\ t = 1, \ldots, n \qquad \text{vs} \qquad H_1 : Y_t | \boldsymbol{P}_t \sim \mu_{1,P_t},\ t = 1, \ldots, n$$

- Not unique, may lead to information loss, but convenient.
- A good choice of Y should have a similar distributional shift for any $\boldsymbol{P}_t$.
- Test distributional difference rather than independence using test score $T_h = \sum_{t=1}^{n} h(Y_t)$ for some score function h. Reject H_0 if T_h is larger than a threshold.

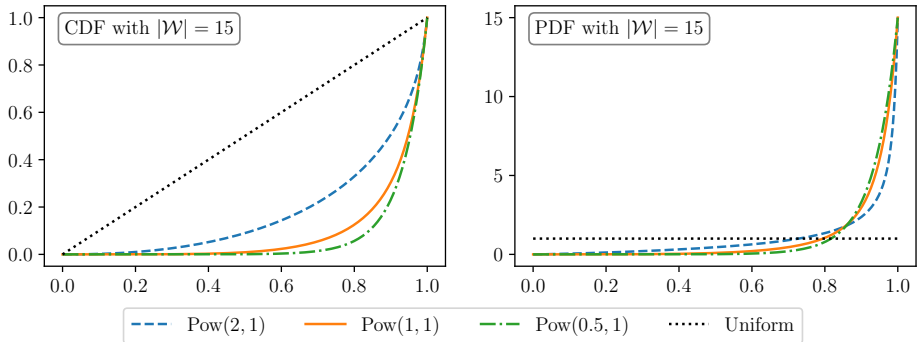
Pseudocode for watermark detection

Algorithm Watermark Detection

1: **Inputs**: a text $w_{\leq n}$, hash function $\mathcal{A}$, secret Key, and significance level α.
2: Simulate n iid samples from the pivotal distributionμ_0.
3: Set $\widehat{q}_n$ as the empirical $(1-\alpha)$-quantile of those null samples, if its theoretical counterpart is hard to compute.
4: Initialize $w_{<1}$ by any prefix.
5: **for** step $t = 1, 2, \cdots, n$ **do**
6: Compute the pseudorandom number: $\zeta_t = \mathcal{A}(w_{<t}, \mathrm{Key})$.
7: Compute the pivotal statistic: $Y_t = Y(w_t, \zeta_t)$.
8: Compute the test score: $T = \mathrm{Score}(Y_{\leq n})$.
9: **Claim**: LLM-generated if $T > \widehat{q}_n$ otherwise human-written.

Pivot for Gumbel-max watermark

- The pivotal statistic is $Y_t^{\text{ars}} = U_{t,w_t}$ given that $\mathcal{S}^{\text{gum}}(\boldsymbol{P}, \zeta) = \arg\max_w \frac{\log \zeta_w}{P_w}$ and $\zeta_t = (U_{t,1}, \ldots, U_{t,K})$.
- Under H_0, $Y_t^{\text{ars}} \overset{iid}{\sim} \mu_0 = \mathcal{U}(0,1)$.
- Under H_1, the CDF of $\mu_{1,\boldsymbol{P}_t}$ is $\mathbb{P}_1(Y_t^{\text{ars}} \leq r | \boldsymbol{P}_t) = \sum_{k=1}^{K} P_{t,k} r^{1/P_{t,k}}$.



Detection for Gumbel-max watermark

Definition (*Default detection for Gumbel-max*)

Aaronson [2023] rejects H_0 if the following $T_{h_{\mathrm{ars}}}$ is larger than a given threshold:

$$T_{h_{\mathrm{ars}}} = \sum_{t=1}^{n} h_{\mathrm{ars}}(Y_t^{\mathrm{ars}}) \quad \text{where} \quad h_{\mathrm{ars}}(y) = -\log(1-y).$$

- Under H_0, $h_{\mathrm{ars}}(Y_t^{\mathrm{ars}}) \overset{iid}{\sim} \mathrm{Exp}(1)$ so that $\mathbb{E}_0[T_{\mathrm{ars}}] = n$.
- Under H_1, $\mathbb{E}_1[T_{\mathrm{ars}}] \geq n + \left(\frac{\pi^2}{6} - 1\right) \sum_{t=1}^{n} \mathbb{E}_1 \mathrm{Ent}(\boldsymbol{P}_t)$ where $\mathrm{Ent}(\boldsymbol{P}_t)$ is Shannon entropy defined by $-\sum_{k=1}^{K} P_{t,k} \log P_{t,k}$.
- Using the same Y_t^{ars}, Fernandez et al. [2023a] finds that $-\log(1-y)$ works better than the variant $\log y$.

Detection for Gumbel-max watermark

Definition (*Default detection for Gumbel-max*)

Aaronson [2023] rejects H_0 if the following $T_{h_{\mathrm{ars}}}$ is larger than a given threshold:

$$T_{h_{\mathrm{ars}}} = \sum_{t=1}^{n} h_{\mathrm{ars}}(Y_t^{\mathrm{ars}}) \quad \text{where} \quad h_{\mathrm{ars}}(y) = -\log(1-y).$$

- Under H_0, $h_{\mathrm{ars}}(Y_t^{\mathrm{ars}}) \overset{iid}{\sim} \mathrm{Exp}(1)$ so that $\mathbb{E}_0[T_{\mathrm{ars}}] = n$.
- Under H_1, $\mathbb{E}_1[T_{\mathrm{ars}}] \geq n + \left(\frac{\pi^2}{6} - 1\right) \sum_{t=1}^{n} \mathbb{E}_1 \mathrm{Ent}(\boldsymbol{P}_t)$ where $\mathrm{Ent}(\boldsymbol{P}_t)$ is Shannon entropy defined by $-\sum_{k=1}^{K} P_{t,k} \log P_{t,k}$.
- Using the same Y_t^{ars}, Fernandez et al. [2023a] finds that $-\log(1-y)$ works better than the variant $\log y$.
- The demo includes totally four detection rules (with other two mentioned latter):

  https://github.com/lx10077/WatermarkFramework/blob/main/watermark_demo.py

Outlines

Motivation from a statistical perspective

Questions

- Find an efficiency measure to rank different detection rules?
- What is the "optimal" score function h?
- Find the optimal detection rule according to the efficiency measure?

Detection efficiency

Fixing Type I error, a detection rule is preferred if it has a higher power.

- However, the comparison depends on the unknown NTP distributions $\boldsymbol{P}_{\leq n}$.

Detection efficiency

Fixing Type I error, a detection rule is preferred if it has a higher power.

- However, the comparison depends on the unknown NTP distributions $\boldsymbol{P}_{\leq n}$.

Minimax viewpoint: the lowest power over all NTP distributions?

- All detection rules are powerless as $Y(w_t, \zeta_t)$ has the same distribution under H_0 and H_1 if $\boldsymbol{P}_t$ has an entry of 1.

Detection efficiency

Fixing Type I error, a detection rule is preferred if it has a higher power.

- However, the comparison depends on the unknown NTP distributions $\boldsymbol{P}_{\leq n}$.

Minimax viewpoint: the lowest power over all NTP distributions?

- All detection rules are powerless as $Y(w_t, \zeta_t)$ has the same distribution under H_0 and H_1 if $\boldsymbol{P}_t$ has an entry of 1.

Class-dependent efficiency

- Select a class $\mathcal{P}$ that is believed to contain all $\boldsymbol{P}_{\leq n}$.
- Evaluate efficiency by the least-favorable power attained over $\mathcal{P}$.
- What is a reasonable class $\mathcal{P}$?

Detection efficiency

Fixing Type I error, a detection rule is preferred if it has a higher power.

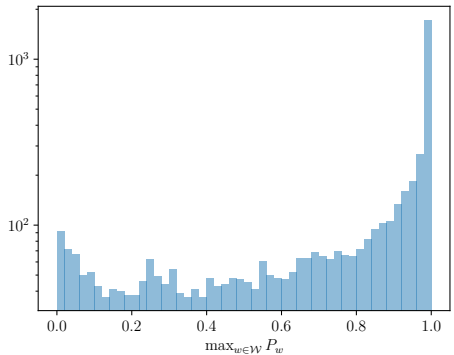
- However, the comparison depends on the unknown NTP distributions $\boldsymbol{P}_{\leq n}$.

Minimax viewpoint: the lowest power over all NTP distributions?

- All detection rules are powerless as $Y(w_t, \zeta_t)$ has the same distribution under H_0 and H_1 if $\boldsymbol{P}_t$ has an entry of 1.

Class-dependent efficiency

- Select a class $\mathcal{P}$ that is believed to contain all $\boldsymbol{P}_{\leq n}$.
- Evaluate efficiency by the least-favorable power attained over $\mathcal{P}$.
- What is a reasonable class $\mathcal{P}$?



Top prob. from ChatGPT-3.5-turbo.

A class of NTP distributions

Δ-regular distribution class:

$$\mathcal{P}_\Delta := \{\boldsymbol{P} = (P_1, \cdots, P_k) : \max_k P_k \leq 1 - \Delta\}.$$

- Chopping off *deterministic* NTP distributions of the form $(0, \ldots, 0, 1, 0, \ldots, 0)$.
- Closely related to the temperature parameter in LLMs.
- Shannon entropy satisfies

$$\mathrm{Ent}(\boldsymbol{P}) = \sum P_w \log \frac{1}{P_w} \geq \sum P_w(1 - P_w) \geq \sum P_w \cdot \Delta = \Delta.$$

Asymptotic class-dependent efficiency

Theorem (*Least-favorable detection efficiency*)

Fixing Type I error in $(0,1)$, the pivot-based test statistic $T_h = \sum h(Y_t)$ satisfies

$$\lim_{n\to\infty} \sup [\text{Type II error}]^{\frac{1}{n}} \leq \exp(-R_{\mathcal{P}}(h)),$$

where $\mathcal{P}$-efficiency rate $R_{\mathcal{P}}(h)$ is defined as

$$R_{\mathcal{P}}(h) = -\inf_{\theta\geq 0} \{\theta \mathbb{E}_0[h(Y)] + \log \phi_{\mathcal{P},h}(\theta)\} \quad \text{with} \quad \phi_{\mathcal{P},h}(\theta) = \sup_{\boldsymbol{P}\in\mathcal{P}} \mathbb{E}_{1,\boldsymbol{P}}\left[e^{-\theta h(Y)}\right].$$

Asymptotic class-dependent efficiency

Theorem (*Least-favorable detection efficiency*)

Fixing Type I error in $(0,1)$, the pivot-based test statistic $T_h = \sum h(Y_t)$ satisfies

$$\limsup_{n\to\infty} [\text{Type II error}]^{\frac{1}{n}} \leq \exp(-R_{\mathcal{P}}(h)),$$

where $\mathcal{P}$-efficiency rate $R_{\mathcal{P}}(h)$ is defined as

$$R_{\mathcal{P}}(h) = -\inf_{\theta\geq 0} \{\theta \mathbb{E}_0[h(Y)] + \log \phi_{\mathcal{P},h}(\theta)\} \quad \text{with} \quad \phi_{\mathcal{P},h}(\theta) = \sup_{\boldsymbol{P}\in\mathcal{P}} \mathbb{E}_{1,\boldsymbol{P}}\left[e^{-\theta h(Y)}\right].$$

- Tight in the minimax sense. Bahadur efficiency when $\mathcal{P}$ is a singleton.
- Monotonicity: $R_{\mathcal{P}_1}(h) \geq R_{\mathcal{P}_2}(h)$ if $\mathcal{P}_1 \subset \mathcal{P}_2$.
- For a mixture from $\mathcal{P}_1$ and $\mathcal{P}_2$ with proportions γ and $1-\gamma$:

$$\limsup_{n\to\infty} \text{Type II error}^{\frac{1}{n}} \leq \exp(-\gamma R_{\mathcal{P}_1}(h) - (1-\gamma) R_{\mathcal{P}_2}(h)).$$

Proof sketch for the asymptotic bound

- For a given score function h, the test rejects H_0 if $\sum_{t=1}^{n} h(Y_t) \geq \gamma_{n,\alpha}$.

Step 1 – calibrating $\gamma_{n,\alpha}$ (Type I)

$$\text{Type I error } = \mathbb{P}_0(T_n \geq \gamma_{n,\alpha}) = \alpha. \implies \frac{\gamma_{n,\alpha}}{n} \xrightarrow[n\to\infty]{} \mathbb{E}_0\, h(Y).$$

Step 2 – bounding Type II by Chernoff bound

$$\text{Type II error } = \mathbb{P}_1\Big(\sum_{t=1}^{n} -h(Y_t) \geq -\gamma_{n,\alpha}\Big) \leq \exp(\gamma_{n,\alpha}\theta) \cdot \exp(n \cdot \log \phi_{\mathcal{P},h}(\theta)).$$

Step 3 – Putting together

$$\begin{aligned}
\limsup_{n\to\infty} [\text{Type II error}]^{1/n} &\leq \limsup_{n\to\infty} \inf_{\theta\geq 0} \exp(\theta\gamma_{n,\alpha}/n) \cdot \exp(\log \phi_{\mathcal{P},h}(\theta)) \\
&\leq \inf_{\theta\geq 0} \limsup_{n\to\infty} \exp(\theta\gamma_{n,\alpha}/n) \cdot \exp(\log \phi_{\mathcal{P},h}(\theta)) \\
&= \inf_{\theta\geq 0} \exp(\theta\mathbb{E}_0 h(Y) + \log \phi_{\mathcal{P},h}(\theta)).
\end{aligned}$$

Optimal score via class-dependent efficiency

Definition (*Optimal score function*)

Finding the optimal score $h^\star = \arg\max_h R_{\mathcal{P}}(h)$ reduces to a minimax problem:

$$\min_h \max_{\boldsymbol{P}\in\mathcal{P}} L(h, \boldsymbol{P}) \quad \text{where} \quad L(h, \boldsymbol{P}) := \mathbb{E}_0[h(Y)] + \log\left(\mathbb{E}_{1,\boldsymbol{P}}\, \mathrm{e}^{-h(Y)}\right).$$

- The minimax problem $\min_h \max_{\boldsymbol{P}\in\mathcal{P}} L(h, \boldsymbol{P})$ is generally not convex-concave.
- Case-by-case analysis is required, but we are often lucky.

How to maximize $R_{\mathcal{P}}(h)$

- Find the saddle point $(\boldsymbol{P}^\star, h^\star)$ that solves the minimax problem:

$$\min_h \max_{\boldsymbol{P}\in\mathcal{P}} L(h, \boldsymbol{P}) = \inf_h \left\{ \mathbb{E}_0[h(Y)] + \sup_{\boldsymbol{P}\in\mathcal{P}} \log\left(\mathbb{E}_1[e^{-h(Y)}|\boldsymbol{P}]\right) \right\}.$$

Theorem (*Saddle point condition*)

If there exists an $\boldsymbol{P}^\star \in \mathcal{P}$ and a score function class $\mathcal{H}$ such that for all $h \in \mathcal{H}$,

$$\sup_{\boldsymbol{P}\in\mathcal{P}} \mathbb{E}_1[e^{-h(Y)}|\boldsymbol{P}] = \mathbb{E}_1[e^{-h(Y)}|\boldsymbol{P}^\star], \qquad (\boldsymbol{P}^\star)$$

$$h^\star := \log \frac{\mathrm{d}\mu_{1,\boldsymbol{P}^\star}}{\mathrm{d}\mu_0} \in \mathcal{H}, \qquad (h^\star)$$

we then have

$$\max_h R_{\mathcal{P}}(h) = L(h^\star, \boldsymbol{P}^\star) = D_{\mathrm{KL}}(\mu_0, \mu_{1,\boldsymbol{P}^\star}),$$

where the maximum is obtained at $h^\star$.

Optimal score for Gumbel-max watermark

Theorem (*Optimal score for Gumbel-max watermark*)

The optimal score that maximizes $h^\star_\Delta := \arg\max_h R_{\mathcal{P}_\Delta}(h)$ is

$$h_{\mathrm{opt},\Delta}(y) = \log \frac{\mathrm{d}\mu_{1,\boldsymbol{P}^\star_\Delta}}{\mathrm{d}\mu_0},$$

where

$$\boldsymbol{P}^\star_\Delta = \Big(\underbrace{1-\Delta,\ldots,1-\Delta}_{\lfloor \frac{1}{1-\Delta} \rfloor \text{ times}}, 1-(1-\Delta)\cdot\left\lfloor \frac{1}{1-\Delta}\right\rfloor, 0, \ldots\Big).$$

Optimal score for Gumbel-max watermark

Theorem (*Optimal score for Gumbel-max watermark*)

The optimal score that maximizes $h^\star_\Delta := \arg\max_h R_{\mathcal{P}_\Delta}(h)$ is

$$h_{\mathrm{opt},\Delta}(y) = \log \frac{\mathrm{d}\mu_{1,\boldsymbol{P}^\star_\Delta}}{\mathrm{d}\mu_0},$$

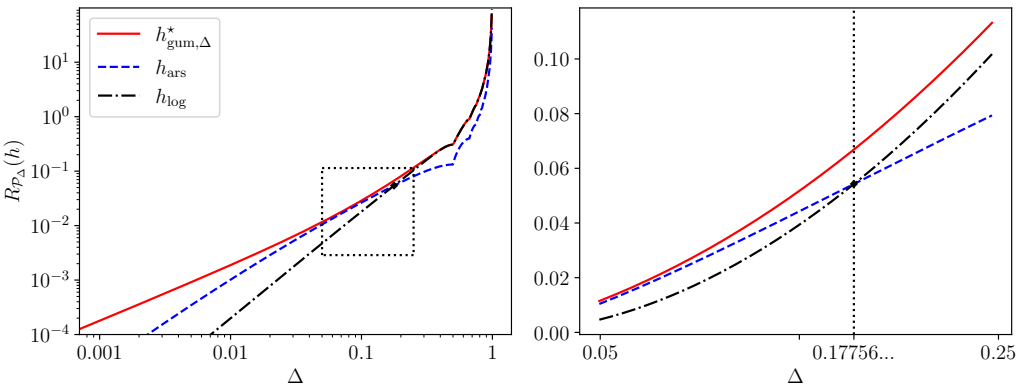
where

$$\boldsymbol{P}^\star_\Delta = \Big(\underbrace{1-\Delta,\ldots,1-\Delta}_{\lfloor \frac{1}{1-\Delta} \rfloor \text{ times}}, 1-(1-\Delta)\cdot\left\lfloor \frac{1}{1-\Delta}\right\rfloor, 0,\ldots\Big).$$

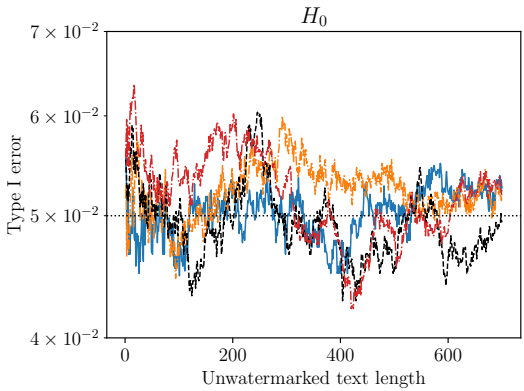
- Key observation: $\mathbb{E}_1[e^{-h(Y)}|\boldsymbol{P}]$ is convex in $\boldsymbol{P}$ for any non-decreasing h.
- $\boldsymbol{P}^\star_\Delta$ is the only extreme point in $\mathcal{P}_\Delta$ up to permutation.
- When $0 < \Delta < 0.5$, $h_{\mathrm{opt},\Delta} = \log\big(y^{\frac{\Delta}{1-\Delta}} + y^{\frac{1-\Delta}{\Delta}}\big)$ as $\boldsymbol{P}^\star_\Delta = (1-\Delta, \Delta, 0, \ldots)$.

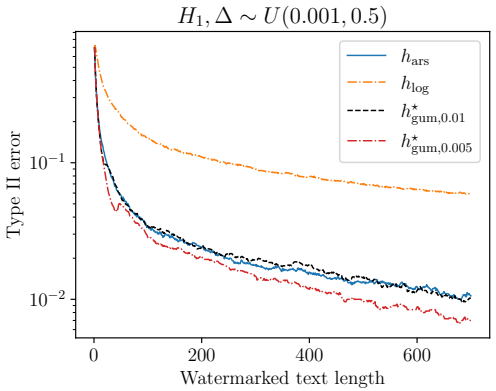
Efficiency plot for Gumbel-max watermark

- Aaronson [2023] proposed $h_{\mathrm{ars}}(y) = -\log(1-y)$.
- Kuditipudi et al. [2024], Fernandez et al. [2023a] proposed $h_{\log}(y) = \log y$.



Simulation results for Gumbel-max watermark





Watermark under text modification

A student might modify the text generated from an LLM, either due to personalization or to try to escape from detection.

Watermark under text modification

A student might modify the text generated from an LLM, either due to personalization or to try to escape from detection.

- To cope with modification, Gumbel-max watermark uses a few tokens to compute pseudorandom numbers

  For example, $\zeta_t = \mathcal{A}(w_{(t-5):(t-1)}, \mathrm{Key})$, using the last 5 tokens.

- A modified token will turn the watermark signals in the next few 5 tokens to noise.

Watermark under text modification

A student might modify the text generated from an LLM, either due to personalization or to try to escape from detection.

- To cope with modification, Gumbel-max watermark uses a few tokens to compute pseudorandom numbers

  For example, $\zeta_t = \mathcal{A}(w_{(t-5):(t-1)}, \mathrm{Key})$, using the last 5 tokens.

- A modified token will turn the watermark signals in the next few 5 tokens to noise.

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{vs} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t$$

Watermark under text modification

A student might modify the text generated from an LLM, either due to personalization or to try to escape from detection.

- To cope with modification, Gumbel-max watermark uses a few tokens to compute pseudorandom numbers

  For example, $\zeta_t = \mathcal{A}(w_{(t-5):(t-1)}, \mathrm{Key})$, using the last 5 tokens.

- A modified token will turn the watermark signals in the next few 5 tokens to noise.

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{vs} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t$$

- $\varepsilon_n \in [0,1]$ denote the fraction of watermark signals.
- When $\varepsilon_n \equiv 1$, it reduced to the full detection setting we considered.

How to solve the mixture detection

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{vs} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1 - \varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t$$

- Difficulties: We know nothing about ε_n or $\boldsymbol{P}_t$.

How to solve the mixture detection

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{vs} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1 - \varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t$$

- Difficulties: We know nothing about ε_n or $\boldsymbol{P}_t$.

Key insight

We know everything about the null H_0 which always assume $Y_{\leq n}$ iid from μ_0.

How to solve the mixture detection

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{vs} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t$$

- Difficulties: We know nothing about ε_n or $\boldsymbol{P}_t$.

Key insight

We know everything about the null H_0 which always assume $Y_{\leq n}$ iid from μ_0.

- Focus to determine whether the observed $Y_1, \ldots, Y_n$ follows μ_0.
- Tr-GoF [Li et al., 2024b] checks the deviation between the empirical CDF of $Y_{\leq n}$ and μ_0 via an f-divergence.
- Too large deviation indicate the existence of watermarked subtexts.

Tr-GoF [Li et al., 2024b]

- The empirical CDF of p-values: $\mathbb{F}_n(r) = \frac{1}{n}\sum_{t=1}^{n} \mathbf{1}_{\mathrm{p}_t \le r}$ where $\mathrm{p}_t = 1 - F_0(Y_t)$.
- Under H_0, we must have $\mathrm{p}_1, \dots, \mathrm{p}_n$ i.i.d. from $\mathcal{U}(0,1)$.
- Tr-GoF rejects H_0 if $\sup_{r\in(0,1)} nK_s(\mathbb{F}_n(r), r)$ is larger than expected for $s \in [-1, 2]$ where

$$K_s(u,v) = \frac{1}{s(1-s)}\left[1 - u^s v^{1-s} - (1-u)^s (1-v)^{1-s}\right].$$

- One can show that K_s is the ϕ_s-divergence between two Bernoulli variables:

$$K_s(u,v) = D_{\phi_s}(\mathrm{Ber}(u) \parallel \mathrm{Ber}(v)) = v\phi_s\left(\frac{u}{v}\right) + (1-v)\phi_s\left(\frac{1-u}{1-v}\right),$$

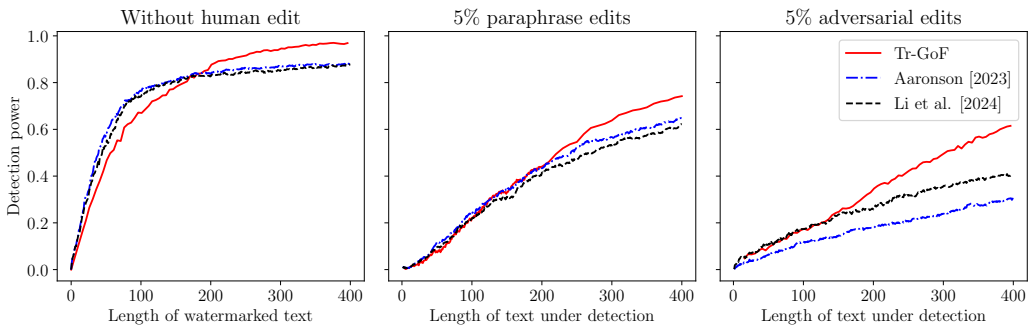
where ϕ_s is a scalar convex function indexed by s:

$$\phi_s(x) = \begin{cases} x\log x - x + 1, & \text{if } s = 1, \\ \frac{1-s+sx-x^s}{s(1-s)}, & \text{if } s \neq 0, 1, \\ -\log x + x - 1, & \text{if } s = 0. \end{cases}$$

- GitHub: `https://github.com/lx10077/TrGoF`.

Robust performance of Tr-GoF

- On C4 news-like dataset [Raffel et al., 2020] and OPT-1.3B model [Zhang et al., 2022] (temperature 0.3).



Connection with general goodness-of-fit (GoF) tests

- In general, GoF tests evaluate whether i.i.d. datas follow μ_0 or μ_1.
- Different GoF tests use different measuress of deviation.
- He et al. [2025] shows that they often improve power and robustness.

Table: Type I errors on human data and Type II errors (averaged over three LLMs) on the C4 dataset for the Gumbel-max watermark. All values are multiplied by 100 for readability.

Temperature	n	Baseline	Tr-GoF	Kui	Kol	And	Cra	Wat	Ney	Chi
0.3	200	18.5	21.0	26.3	19.5	**15.5**	21.2	36.8	19.7	18.5
	400	15.1	5.7	4.7	4.7	4.9	8.4	10.7	8.0	**2.9**
0.7	200	0.6	**0.3**	0.5	0.6	0.5	0.7	0.9	0.5	**0.3**
	400	0.7	**0.2**	**0.2**	0.3	**0.2**	0.4	0.4	**0.2**	**0.2**
Type I		–	0.4	0.9	1.5	0.6	0.7	1.2	1.1	0.9

Table: Goodness-of-fit tests and their sources.

Test name	**Reference**
Tr-GoF test (`Tr-GoF`)	[Li et al., 2024b]
Kuiper's test (`Kui`)	[Kuiper, 1960]
Kolmogorov–Smirnov test (`Kol`)	[Smirnov, 1939]
Anderson–Darling test (`And`)	[Anderson and Darling, 1952]
Cramér–von Mises test (`Cra`)	[Cramér, 1928]
Watson's test (`Wat`)	[Watson, 1961]
Neyman's smooth test (`Ney`)	[Neyman, 1937]
Chi-squared test (`Chi`)	[Pearson, 1900]

Why the Tr-GoF test performs so well?

A question

Why the Tr-GoF test performs so well in the watermark detection problem?

- We focus on the Gumbel-max watermark. Similar analysis could be paralleled to other watermarks.

Why the Tr-GoF test performs so well?

A question

Why the Tr-GoF test performs so well in the watermark detection problem?

- We focus on the Gumbel-max watermark. Similar analysis could be paralleled to other watermarks.

High-level answers

The Tr-GoF test achieves optimal robustness in two senses:

1. Optimal detection boundary in a decaying watermark-signal case.
2. Optimal detection efficiency rate in a constant corruption case.

!!! No knowledge about the fraction ε_n and NTP distributions.

When the robust detection is possible?

Optimal detection boundary

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{versus} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1 - \varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t.$$

When the robust detection is possible?

Optimal detection boundary

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{versus} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t.$$

Definition (*A difficult case*)

We consider an extreme case where

- $\varepsilon_n \asymp n^{-p}$ for all t and $p \in (0,1]$.
- $1 - \max_{w \in \mathcal{W}} \boldsymbol{P}_{t,w} = \Delta_n$ for all t with $\Delta_n \asymp n^{-q}$ and $q \in (0,1)$.

When the robust detection is possible?

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{versus} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t.$$

Definition (*A difficult case*)

We consider an extreme case where

- $\varepsilon_n \asymp n^{-p}$ for all t and $p \in (0,1]$.
- $1 - \max_{w \in \mathcal{W}} \boldsymbol{P}_{t,w} = \Delta_n$ for all t with $\Delta_n \asymp n^{-q}$ and $q \in (0,1)$.

- Motivated by sparse detection problem [Donoho and Jin, 2004, 2015].
- If $\varepsilon_n = 0$ or $1 - \max_{w \in \mathcal{W}} \boldsymbol{P}_{t,w} = 0$, $(1-\varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} = \mu_0$, i.e., H_0 merges with H_1^{mix}.

When the robust detection is possible?

Theorem (*Optimal detection boundary on the* (p, q)*-plane*)

- If $q + 2p > 1$, H_0 and H_1^{m} merge asymptotically. For any test, the sum of Type I and Type II error probabilities is 1 as $n \to \infty$.
- If $q + 2p < 1$, H_0 and H_1^{m} separate asymptotically. Furthermore, for the likelihood-ratio test that rejects H_0 if the log-likelihood ratio is positive, the sum of Type I and Type II error probabilities tends to 0 as $n \to \infty$.

When the robust detection is possible?

Theorem (*Optimal detection boundary on the* (p, q)*-plane*)

- If $q + 2p > 1$, H_0 and H_1^{m} merge asymptotically. For any test, the sum of Type I and Type II error probabilities is 1 as $n \to \infty$.
- If $q + 2p < 1$, H_0 and H_1^{m} separate asymptotically. Furthermore, for the likelihood-ratio test that rejects H_0 if the log-likelihood ratio is positive, the sum of Type I and Type II error probabilities tends to 0 as $n \to \infty$.

$\Longrightarrow$ Robust detection is impossible for small watermark signal, i.e., $q + 2p > 1$.

When the robust detection is possible?

Optimal detection boundary

Theorem (*Optimal detection boundary on the* (p, q)*-plane*)

- If $q + 2p > 1$, H_0 and H_1^{m} merge asymptotically. For any test, the sum of Type I and Type II error probabilities is 1 as $n \to \infty$.
- If $q + 2p < 1$, H_0 and H_1^{m} separate asymptotically. Furthermore, for the likelihood-ratio test that rejects H_0 if the log-likelihood ratio is positive, the sum of Type I and Type II error probabilities tends to 0 as $n \to \infty$.

$\Longrightarrow$ Robust detection is impossible for small watermark signal, i.e., $q + 2p > 1$.

$\Longrightarrow$ With sufficient watermark signal, detection is possible with the likelihood-ratio test an optimal rule, i.e., $q + 2p < 1$.

!!! The likelihood-ratio test is impractical as it needs to know $\boldsymbol{P}_t$'s and ε_n.

Adaptive optimality

Target

An ideal optimal detection method should work as long as $q + 2p < 1$ and don't requires the knowledge of $\boldsymbol{P}_t$'s and ε_n.

Adaptive optimality

Target

An ideal optimal detection method should work as long as $q + 2p < 1$ and don't requires the knowledge of $\boldsymbol{P}_t$'s and ε_n.

Our finding

The GoF test achieves this optimal detection boundary.

Adaptive optimality

Target

An ideal optimal detection method should work as long as $q + 2p < 1$ and don't requires the knowledge of $\boldsymbol{P}_t$'s and ε_n.

Our finding

The GoF test achieves this optimal detection boundary.

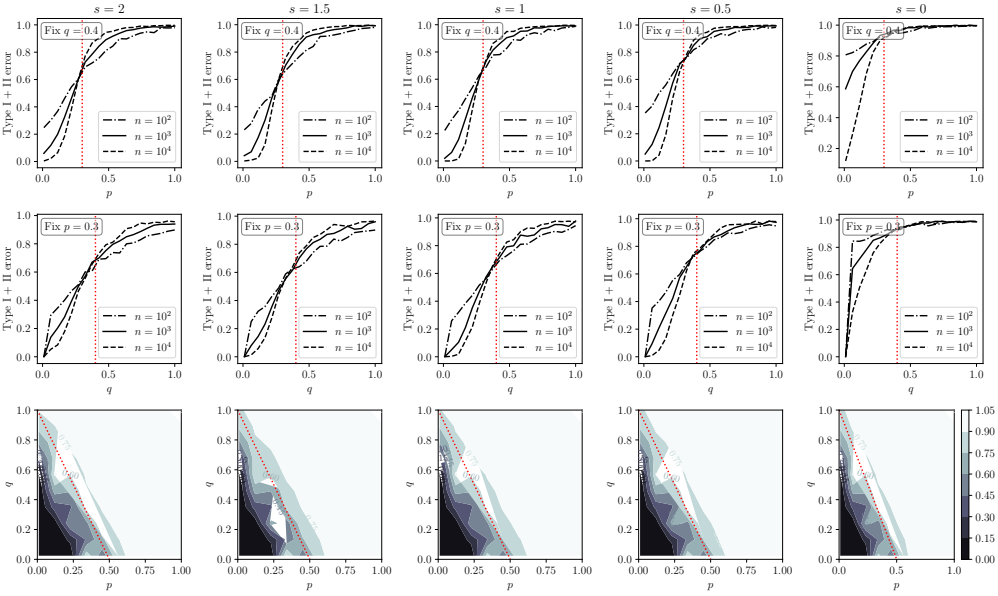
Theorem (*Adaptive optimality*)

The Type I and II errors of the Tr-GoF test $\to 0$ if $n \to \infty$ as long as $q + 2p < 1$.

- Optimal adaptivity without any prior knowledge.

Empirical detection boundaries of Tr-GoF

Optimal detection boundary



Suboptimality of sum-based tests

- Consider the sum-based test that rejects H_0 if

$$\sum_{t=1}^{n} h(Y_t^{\mathrm{ars}}) \geq n \cdot \mathbb{E}_0[h(Y^{\mathrm{ars}})] + \Theta(1) \cdot n^{\frac{1}{2}} \cdot \mathrm{poly}(\log n).$$

Theorem (*Suboptimality of sum-based tests*)

When $\varepsilon < 1$, the detection boundary for general (Δ, ε)-agnostic sum-based tests is $q + p = 1/2$ (which include $h \in \{h_{\mathrm{ars}}, h_{\log}, h_{\mathrm{opt},\Delta}\}$).

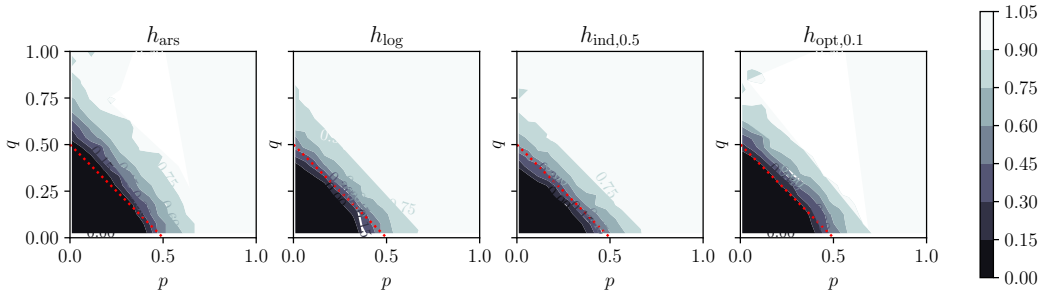
Suboptimality of sum-based tests

- Consider the sum-based test that rejects H_0 if

$$\sum_{t=1}^{n} h(Y_t^{\mathrm{ars}}) \geq n \cdot \mathbb{E}_0[h(Y^{\mathrm{ars}})] + \Theta(1) \cdot n^{\frac{1}{2}} \cdot \mathrm{poly}(\log n).$$

Theorem (*Suboptimality of sum-based tests*)

When $\varepsilon < 1$, the detection boundary for general (Δ, ε)-agnostic sum-based tests is $q + p = 1/2$ (which include $h \in \{h_{\mathrm{ars}}, h_{\log}, h_{\mathrm{opt},\Delta}\}$).



What about constant corruption?

- The optimal detection boundary cares about the diminishing region where the watermark signal decays with the text length n.
- Practical settings meet with the constant corruption case, i.e., $\varepsilon_n \equiv \varepsilon$.
- The problem is detectable because $p = q = 0$ (within $q + 2p < 1$).

What about constant corruption?

- The optimal detection boundary cares about the diminishing region where the watermark signal decays with the text length n.
- Practical settings meet with the constant corruption case, i.e., $\varepsilon_n \equiv \varepsilon$.
- The problem is detectable because $p = q = 0$ (within $q + 2p < 1$).
- Recall $\mathcal{P}$-efficiency: the rate of exponential decrease in Type II errors for a fixed significance level α and the worst-case alternative within a belief set $\mathcal{P}$.

Definition ($\mathcal{P}$-*efficiency [Li et al., 2025a], revisited*)

Let $\gamma_{n,\alpha}$ satisfy $\mathbb{P}_0(S_n \geq \gamma_{n,\alpha}) = \alpha$ for $n \geq 1$. For a given belief set $\mathcal{P}$, we define the following limit (if exists) as the $\mathcal{P}$-efficiency of S_n and denote it by $R_{\mathcal{P}}(S_n)$:

$$\lim_{n\to\infty} \sup_{\boldsymbol{P}_t \in \mathcal{P}, \forall t \in [n]} \frac{1}{n} \log \mathbb{P}_1(S_n \leq \gamma_{n,\alpha}) = -R_{\mathcal{P}}(S_n).$$

What about constant corruption?

Theorem (*Optimal $\mathcal{P}_\Delta$-efficiency*)

Let $s \in (0,1)$, $\varepsilon_n \equiv \varepsilon \in (0,1]$ and $\Delta_n \equiv \Delta \in (0,1)$.

$$R_{\mathcal{P}_\Delta}(\text{any detection rule}) \leq D_{\mathrm{KL}}(\mu_0, (1-\varepsilon)\mu_0 + \varepsilon\mu_{1,\boldsymbol{P}^\star_\Delta}) \leq R_{\mathcal{P}_\Delta}(\mathrm{Tr}-\mathrm{GoF})$$

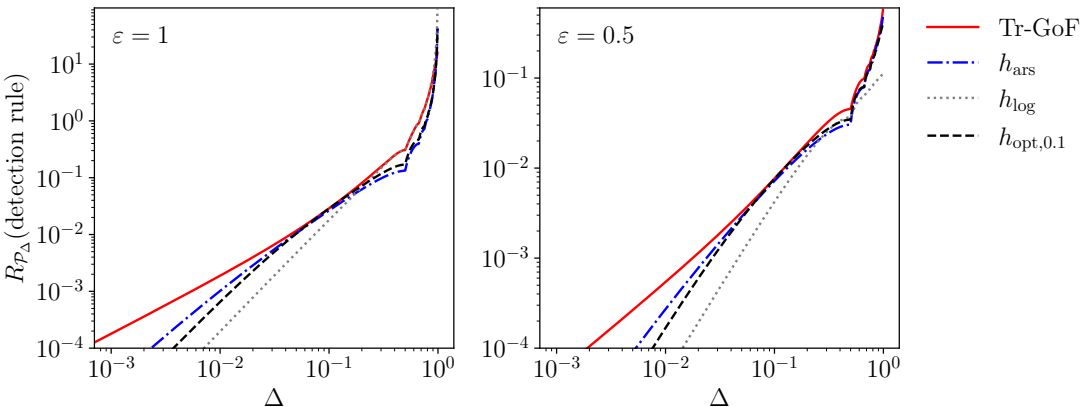
where $\boldsymbol{P}^\star_\Delta$ is the least-favorable NTP distribution defined by

$$\boldsymbol{P}^\star_\Delta = \Big(\underbrace{1-\Delta, \dots, 1-\Delta}_{\lfloor \frac{1}{1-\Delta} \rfloor \text{ times}}, 1-(1-\Delta)\cdot\Big\lfloor\frac{1}{1-\Delta}\Big\rfloor, 0, \dots\Big).$$

- Upper and lower bounds.
- When $\varepsilon = 1$, this rate is obtained by the sum-based test defined by $h_{\mathrm{opt},\Delta}$.
- Optimal efficiency without any prior knowledge.

Theoretical $\mathcal{P}_\Delta$-efficiency comparison

Optimal detection efficiency



Proportion estimation

Hypothesis testing under constant mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{versus} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon)\mu_0 + \varepsilon \mu_{1,\boldsymbol{P}_t} \ \forall t.$$

- Once we have confirmed that $w_{\leq n}$ was generated by the LLM (i.e. reject H_1^{mix}), how many were produced by the model?
- **Application**: Measure LLMs' intellectual contribution.

Proportion estimation

Hypothesis testing under constant mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{versus} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon)\mu_0 + \varepsilon \mu_{1,\boldsymbol{P}_t} \ \forall t.$$

- Once we have confirmed that $w_{\leq n}$ was generated by the LLM (i.e. reject H_1^{mix}), how many were produced by the model?
- **Application**: Measure LLMs' intellectual contribution.

Definition (*Proportion estimation under constant mixtures [Li et al., 2025b]*)

Given independent data $Y_t \sim (1-\varepsilon)F_0 + \varepsilon F_{\boldsymbol{P}_t}$ for all t, how to estimate ε accurately?

Proportion estimation

Hypothesis testing under constant mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{versus} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon)\mu_0 + \varepsilon \mu_{1,\boldsymbol{P}_t} \ \forall t.$$

- Once we have confirmed that $w_{\leq n}$ was generated by the LLM (i.e. reject H_1^{mix}), how many were produced by the model?
- **Application**: Measure LLMs' intellectual contribution.

Definition (*Proportion estimation under constant mixtures [Li et al., 2025b]*)

Given independent data $Y_t \sim (1-\varepsilon)F_0 + \varepsilon F_{\boldsymbol{P}_t}$ for all t, how to estimate ε accurately?

Take-away

ε is not identifiable under every watermarking scheme—not estimable for the green–red list, yet estimable for Gumbel-max.

When ε is identifiable?

Definition (*Proportion estimation under constant mixtures [Li et al., 2025b]*)

Given independent data $Y_t \sim (1-\varepsilon)F_0 + \varepsilon F_{\boldsymbol{P}_t}$ for all t, how to estimate ε accurately?

When ε is identifiable?

Definition (*Proportion estimation under constant mixtures [Li et al., 2025b]*)

Given independent data $Y_t \sim (1-\varepsilon)F_0 + \varepsilon F_{\boldsymbol{P}_t}$ for all t, how to estimate ε accurately?

- ε is not identifiable for green-red list watermark.

Lemma

If $Y_{1:n}$ are i.i.d. from the binary mixture $(1-\varepsilon)\mathrm{Ber}(\gamma) + \varepsilon\mathrm{Ber}(\mu)$ where both ε and μ are unknown with γ known, ε is not identifiable (as $Y_t \overset{iid}{\sim} \mathrm{Ber}((1-\varepsilon)\gamma) + \varepsilon\mu)$).

When ε is identifiable?

Definition (*Proportion estimation under constant mixtures [Li et al., 2025b]*)

Given independent data $Y_t \sim (1-\varepsilon)F_0 + \varepsilon F_{\boldsymbol{P}_t}$ for all t, how to estimate ε accurately?

- ε is not identifiable for green-red list watermark.

Lemma

If $Y_{1:n}$ are i.i.d. from the binary mixture $(1-\varepsilon)\mathrm{Ber}(\gamma) + \varepsilon\mathrm{Ber}(\mu)$ where both ε and μ are unknown with γ known, ε is not identifiable (as $Y_t \overset{iid}{\sim} \mathrm{Ber}((1-\varepsilon)\gamma) + \varepsilon\mu)$).

ε is identifiable for Gumbel-max watermark (and other wm with continuous Y).

Lemma

If $Y_t \sim (1-\varepsilon)F_0 + \varepsilon F_{\boldsymbol{P}_t}$ and $\lim_{x\to 0} \frac{\bar{F}_{\boldsymbol{P}}(x)}{F_0(x)} = 0$, then ε is identifiable (as $\varepsilon = 1 - \lim_{x\to 0} \frac{\bar{F}(x)}{F_0(x)}$ is well-defined).

How to estimate ε when possible?

Some notations

$\bar{F}(y) = (1-\varepsilon)F_0(y) + \varepsilon \bar{F}_{\boldsymbol{P}}(y)$ for all y,

- $\bar{F}(y) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{P}(Y_t \leq y)$
- $F_0(y) = \mu_0(Y \leq y)$
- $\bar{F}_{\boldsymbol{P}}(y) = \frac{1}{n}\sum_{t=1}^{n} \mu_{1,\boldsymbol{P}_t}(Y \leq y)$

How to estimate ε when possible?

Some notations

$\bar{F}(y) = (1-\varepsilon)F_0(y) + \varepsilon \bar{F}_{\boldsymbol{P}}(y)$ for all y,

- $\bar{F}(y) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{P}(Y_t \leq y)$
- $F_0(y) = \mu_0(Y \leq y)$
- $\bar{F}_{\boldsymbol{P}}(y) = \frac{1}{n}\sum_{t=1}^{n} \mu_{1,\boldsymbol{P}_t}(Y \leq y)$

Key idea: "Moment" matching

For any weight function $v : \mathbb{R} \mapsto \mathbb{R}$,

$$\varepsilon = \frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}.$$

How to estimate ε when possible?

Some notations

$\bar{F}(y) = (1-\varepsilon)F_0(y) + \varepsilon \bar{F}_{\boldsymbol{P}}(y)$ for all y,

- $\bar{F}(y) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{P}(Y_t \leq y)$
- $F_0(y) = \mu_0(Y \leq y)$
- $\bar{F}_{\boldsymbol{P}}(y) = \frac{1}{n}\sum_{t=1}^{n} \mu_{1,\boldsymbol{P}_t}(Y \leq y)$

Key idea: "Moment" matching

For any weight function $v : \mathbb{R} \mapsto \mathbb{R}$,

$$\varepsilon = \frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}.$$

Two difficulties: **(1)** no access to $\bar{F}_{\boldsymbol{P}}$ and **(2)** which v to use.

How to estimate ε when possible?

Some notations

$\bar{F}(y) = (1-\varepsilon)F_0(y) + \varepsilon \bar{F}_{\boldsymbol{P}}(y)$ for all y,

- $\bar{F}(y) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{P}(Y_t \leq y)$
- $F_0(y) = \mu_0(Y \leq y)$
- $\bar{F}_{\boldsymbol{P}}(y) = \frac{1}{n}\sum_{t=1}^{n} \mu_{1,\boldsymbol{P}_t}(Y \leq y)$

Key idea: "Moment" matching

For any weight function $v : \mathbb{R} \mapsto \mathbb{R}$,

$$\varepsilon = \frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}.$$

Two difficulties: **(1)** no access to $\bar{F}_{\boldsymbol{P}}$ and **(2)** which v to use.

- **Estimate** $\bar{F}_{\boldsymbol{P}}$: collect water-marked outputs from similar LLMs $\Rightarrow$ empirical $\widehat{F}_{\boldsymbol{P}}$.
- **Choose** v: set heuristically *or* optimize against a clear performance criterion.
- GitHub: `https://github.com/lx10077/WatermarkProportion`.

Proportion estimators

Key observation

For any weight function $v : \mathbb{R} \mapsto \mathbb{R}$,

$$\varepsilon = \frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}.$$

Collected data

- Let $\widehat{F}$ denote the empirical CDF of observed $Y_{\leq n}$.
- Let $\widehat{F}_{\boldsymbol{P}}$ approximate the alternative CDF $\bar{F}_{\boldsymbol{P}}$ (no accuracy guarantees).

Proportion estimators

Key observation

For any weight function $v : \mathbb{R} \mapsto \mathbb{R}$,

$$\varepsilon = \frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}.$$

Collected data

- Let $\widehat{F}$ denote the empirical CDF of observed $Y_{\leq n}$.
- Let $\widehat{F}_{\boldsymbol{P}}$ approximate the alternative CDF $\bar{F}_{\boldsymbol{P}}$ (no accuracy guarantees).

Way 1: Ignore $\bar{F}_{\boldsymbol{P}}$ & use indicator v

$$\widehat{\varepsilon}_{\mathrm{ini}}(\delta) = 1 - \frac{\widehat{F}(\delta)}{F_0(\delta)}.$$

Proportion estimators

Key observation

For any weight function $v : \mathbb{R} \mapsto \mathbb{R}$,

$$\varepsilon = \frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}.$$

Collected data

- Let $\widehat{F}$ denote the empirical CDF of observed $Y_{\leq n}$.
- Let $\widehat{F}_{\boldsymbol{P}}$ approximate the alternative CDF $\bar{F}_{\boldsymbol{P}}$ (no accuracy guarantees).

Way 1: Ignore $\bar{F}_{\boldsymbol{P}}$ & use indicator v

$$\widehat{\varepsilon}_{\mathrm{ini}}(\delta) = 1 - \frac{\widehat{F}(\delta)}{F_0(\delta)}.$$

Way 2: Use $\widehat{F}_{\boldsymbol{P}}$ & indicator v

$$\widehat{\varepsilon}_{\mathrm{rfn}}(\delta) = \frac{F_0(\delta) - \widehat{F}(\delta)}{F_0(\delta) - \widehat{F}_{\boldsymbol{P}}(\delta)}.$$

Proportion estimators

Key observation

For any weight function $v : \mathbb{R} \mapsto \mathbb{R}$,

$$\varepsilon = \frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}.$$

Collected data

- Let $\widehat{F}$ denote the empirical CDF of observed $Y_{\leq n}$.
- Let $\widehat{F}_{\boldsymbol{P}}$ approximate the alternative CDF $\bar{F}_{\boldsymbol{P}}$ (no accuracy guarantees).

Way 1: Ignore $\bar{F}_{\boldsymbol{P}}$ & use indicator v

$$\widehat{\varepsilon}_{\mathrm{ini}}(\delta) = 1 - \frac{\widehat{F}(\delta)}{F_0(\delta)}.$$

Way 2: Use $\widehat{F}_{\boldsymbol{P}}$ & indicator v

$$\widehat{\varepsilon}_{\mathrm{rfn}}(\delta) = \frac{F_0(\delta) - \widehat{F}(\delta)}{F_0(\delta) - \widehat{F}_{\boldsymbol{P}}(\delta)}.$$

Way 3: Use $\widehat{F}_{\boldsymbol{P}}$ & optimal v

Fixed point of $\widehat{\varepsilon}_{\mathrm{opt}} = \widehat{\mathcal{T}}(\widehat{\varepsilon}_{\mathrm{opt}})$ where

$$\widehat{\mathcal{T}}(\varepsilon) = \frac{\int \widehat{v}_{\mathrm{opt}}(\varepsilon, y) \left[\mathrm{d}F_0(y) - \mathrm{d}\widehat{F}(y)\right]}{\int \widehat{v}_{\mathrm{opt}}(\varepsilon, y) \left[\mathrm{d}F_0(y) - \mathrm{d}\widehat{F}_{\boldsymbol{P}}(y)\right]}$$

$$\widehat{v}_{\mathrm{opt}}(\varepsilon, y) = \frac{1 - \widehat{g}(y)}{(1-\varepsilon) + \varepsilon \widehat{g}(y)}, \widehat{g}(y) = \frac{\mathrm{d}\widehat{F}_{\boldsymbol{P}}(y)}{\mathrm{d}F_0(y)}.$$

Why optimal?

Lemma (*Optimal estimator variance*)

If $\widehat{F}_{\boldsymbol{P}} = \bar{F}_{\boldsymbol{P}}$, it follows that

$$\mathrm{Var}\left(\frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\widehat{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}\right) \leq \frac{\mathrm{Var}_{\bar{F}}(v)}{n(\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v])^2}$$

Why optimal?

Lemma (*Optimal estimator variance*)

If $\widehat{F}_{\boldsymbol{P}} = \bar{F}_{\boldsymbol{P}}$, it follows that

$$\operatorname{Var}\left(\frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\widehat{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}\right) \leq \frac{\operatorname{Var}_{\bar{F}}(v)}{n(\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v])^2}$$

Lemma (*Optimal weight function*)

$$\min_v \frac{\operatorname{Var}_{\bar{F}}(v)}{[\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]]^2} = \left[\int \frac{[1 - g(x)]^2}{(1-\varepsilon) + \varepsilon g(x)} \mathrm{d}F_0(x)\right]^{-1}$$

where the optimal solution (up to constant factors) is

$$v_{\mathrm{opt}}(x) = \frac{1 - g(x)}{(1-\varepsilon) + \varepsilon g(x)}, \text{ with } g(x) = \frac{\mathrm{d}\bar{F}_{\boldsymbol{P}}(x)}{\mathrm{d}F_0(x)}.$$

Why optimal?

Lemma (*Optimal weight function*)

$$\min_{v} \frac{\mathrm{Var}_{\bar{F}}(v)}{[\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]]^2} = \left[\int \frac{[1-g(x)]^2}{(1-\varepsilon)+\varepsilon g(x)} \mathrm{d}F_0(x)\right]^{-1} =: [\tau_n^\star]^2,$$

where the optimal solution (up to constant factors) is

$$v_{\mathrm{opt}}(x) = \frac{1-g(x)}{(1-\varepsilon)+\varepsilon g(x)}, \text{ with } g(x) = \frac{\mathrm{d}\bar{F}_{\boldsymbol{P}}(x)}{\mathrm{d}F_0(x)}.$$

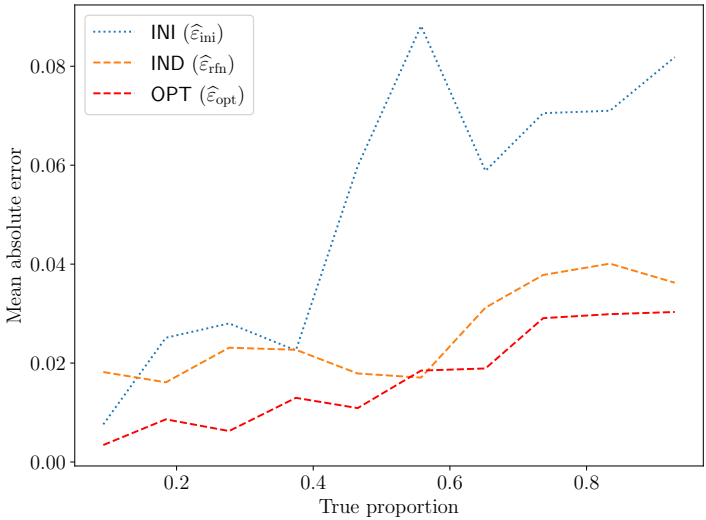
Theorem (*Minimax optimality*)

$\widehat{\varepsilon}_{\mathrm{opt}}$ is the minimax optimal estimator up to the estimation error in $\widehat{F}_{\boldsymbol{P}}$, i.e.,

$$\mathbb{E}|\widehat{\varepsilon}_{\mathrm{opt}} - \varepsilon| \lesssim \frac{\tau_n^\star + o(1)}{\sqrt{n}} + \text{estimation errors due to } \widehat{F}_{\boldsymbol{P}}$$

Empirical performance [Li et al., 2025b]

- On `arXiv` dataset [Cohan et al., 2018] and OPT-13B model [Zhang et al., 2022] (temperature 1.0).



Watermark localization

- Given that an ε fraction of tokens is watermarked, which specific tokens bear the watermark?

Watermark localization

- Given that an ε fraction of tokens is watermarked, which specific tokens bear the watermark?

Definition (*Watermark localization under mixtures*)

Given independent data $Y_t \sim (1-\eta_t)\mu_0 + \eta_t \mu_{1,\boldsymbol{P}_t}$ forall t, how to estimate the binary process $\{\eta_t\}$ accurately?

- In proportion estimation, we assume $\mathbb{E}[\eta_t] \equiv \varepsilon$.
- **Applications**: Authorship classification, mix-source data cleaning...

Watermark localization

- Given that an ε fraction of tokens is watermarked, which specific tokens bear the watermark?

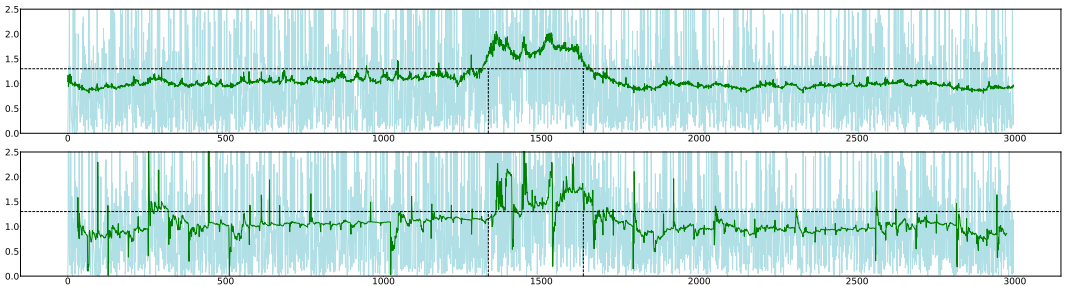
Definition (*Watermark localization under mixtures*)

Given independent data $Y_t \sim (1-\eta_t)\mu_0 + \eta_t \mu_{1,\boldsymbol{P}_t}$ forall t, how to estimate the binary process $\{\eta_t\}$ accurately?

- In proportion estimation, we assume $\mathbb{E}[\eta_t] \equiv \varepsilon$.
- **Applications**: Authorship classification, mix-source data cleaning...
- Localization is harder than proportion estimation [Cai and Sun, 2017].
- Existing methods rate each token (or span) and flag those with high scores as watermarked [Zhao et al., 2024c, Li et al., 2024a].

Watermark localization: AOL [Zhao et al., 2024c]

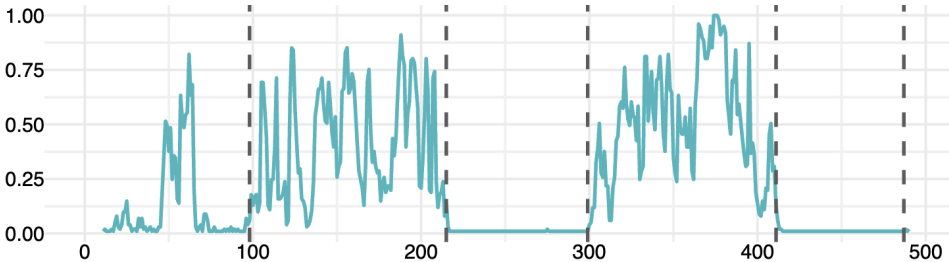
- Apply an online learning algorithm to estimate $\{\mathbb{E}[\eta_t]\}$ for a given text.
- Label a token as watermarked if its score exceeds a threshold.



Estimated token scores v.s. token indexes on Gumbel-max (threshold 1.3 and length $3,000$).

Watermark localization: SeedBS [Li et al., 2024a]

- Treat localization as a change-point problem on the token-wise p-value sequence.
- Mark an interval as watermarked when its score is too large.
- All candidate intervals are scanned via binary segmentation.

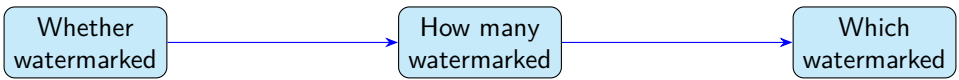


p-value v.s. token indexes on Gumbel-max (threshold 0.005 and length 500).

Outlines

Concluding remarks

- Watermarking enhances content traceability and integrity.
- Applicable to various data modalities: text, images, audio, proteins, etc.
- Each token is coupled with tractable pseudorandom numbers for embedding.
- Detection relies on identifying this coupling via pivotal statistics.
- Detection performance is measured through class-dependent efficiency.
- Statistical problems arise in practical watermarking applications:



- Have a try to watermark!

  https://github.com/lx10077/WatermarkFramework/blob/main/watermark_demo.py

Potential directions

- **Detection beyond (scalar) pivotal statistics.**
  Are there more effective functions for detecting shifts in the joint dist. of (w, ζ)*?*
- **Optimal watermarking:** capacity, detectability, and stealth trade-offs.
  What is the optimal bit-rate under a fidelity constraint?
- **Online watermarking and real-time detection.**
  Can we detect machine generation on the fly under data shift?
- **Multiple or overlapping watermarks.**
  If multiple watermarks coexist, how can we separate and attribute them?
- **Robustness to paraphrasing and translation.**
  Can we design paraphrase-invariant watermarking?
- **Post-generation watermarking.**
  Is it possible to embed watermarks into an already generated text?
- ...

References I

S. Aaronson. Watermarking of large language models. https://simons.berkeley.edu/talks/scott-aaronson-ut-austin-openai-2023-08-17, August 2023.

P. Abdalla and R. Vershynin. LLM watermarking using mixtures and statistical-to-computational gaps. *arXiv preprint arXiv:2505.01484*, 2025.

T. W. Anderson and D. A. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The annals of mathematical statistics*, pages 193–212, 1952.

G. Bao, Y. Zhao, Z. Teng, L. Yang, and Y. Zhang. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*, 2023.

B. Barak. An intensive introduction to cryptography, lectures notes for Harvard CS 127. https://intensecrypto.org/public/index.html, Fall 2021.

T. T. Cai and W. Sun. Optimal screening and discovery of sparse signals with applications to multistage high throughput studies. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(1): 197–223, 2017.

Y. Chen, Z. Hu, Y. Wu, R. Chen, Y. Jin, M. Zhan, C. Xie, W. Chen, and H. Huang. Enhancing privacy in biosecurity with watermarked protein design. *Bioinformatics*, page btaf141, 2025.

M. Christ, S. Gunn, and O. Zamir. Undetectable watermarks for language models. In *Conference on Learning Theory*, pages 1125–1139. PMLR, 2024.

A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 2, pages 615–621, 2018.

References II

H. Cramér. On the composition of elementary errors: First paper: Mathematical deductions. *Scandinavian Actuarial Journal*, 1928(1):13–74, 1928.

S. Dathathri, A. See, S. Ghaisas, P.-S. Huang, R. McAdam, J. Welbl, V. Bachani, A. Kaskasoli, R. Stanforth, T. Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634 (8035):818–823, 2024.

D. Donoho and J. Jin. Higher criticism for detecting sparse heterogeneous mixtures. *The Annals of Statistics*, 32(3):962–994, 2004.

D. Donoho and J. Jin. Higher criticism for large-scale inference, especially for rare and weak effects. *Statistical science*, 30(1):1–25, 2015.

P. Fernandez, A. Chaffin, K. Tit, V. Chappelier, and T. Furon. Three bricks to consolidate watermarks for large language models. *arXiv preprint arXiv:2308.00113*, 2023a.

P. Fernandez, G. Couairon, H. Jégou, M. Douze, and T. Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22466–22477, 2023b.

J. Fu, X. Zhao, R. Yang, Y. Zhang, J. Chen, and Y. Xiao. GumbelSoft: Diversified language model watermarking via the GumbelMax-trick. *arXiv preprint arXiv:2402.12948*, 2024.

E. Giboulot and F. Teddy. WaterMax: Breaking the LLM watermark detectability-robustness-quality trade-off. In *Advances in Neural Information Processing Systems*, volume 37, pages 18848–18881, 2024.

References III

T. Gloaguen, N. Jovanović, R. Staab, and M. Vechev. Towards watermarking of open-source LLMs. *arXiv preprint arXiv:2502.10525*, 2025.

GPTZero. GPTZero: More than an AI detector preserve what's human. `https://gptzero.me/`, 2023.

E. J. Gumbel. *Statistical theory of extreme values and some practical applications: A series of lectures*, volume 33. US Government Printing Office, 1948.

W. He, X. Li, T. Shang, L. Shen, W. J. Su, and Q. Long. On the empirical power of goodness-of-fit tests in watermark detection. *To appear on arXiv*, 2025. Manuscript in preparation.

Z. Hu, L. Chen, X. Wu, Y. Wu, H. Zhang, and H. Huang. Unbiased watermark for large language models. In *International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=uWVC5FVidc`.

W. Ji, W. Yuan, E. Getzen, K. Cho, M. I. Jordan, S. Mei, J. E. Weston, W. J. Su, J. Xu, and L. Zhang. An overview of large language models for statisticians. *arXiv preprint arXiv:2502.17814*, 2025.

J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, volume 202, pages 17061–17084, 2023.

J. Kirchenbauer, J. Geiping, Y. Wen, M. Shu, K. Saifullah, K. Kong, K. Fernando, A. Saha, M. Goldblum, and T. Goldstein. On the reliability of watermarks for large language models. In *International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=DEJIDCmW0z`.

References IV

R. Kuditipudi, J. Thickstun, T. Hashimoto, and P. Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=FpaCL1MO2C.

N. H. Kuiper. Tests concerning random points on a circle. In *Nederl. Akad. Wetensch. Proc. Ser. A*, volume 63, pages 38–47, 1960.

X. Li, G. Li, and X. Zhang. Segmenting watermarked texts from language models. In *Neural Information Processing Systems*, 2024a.

X. Li, F. Ruan, H. Wang, Q. Long, and W. J. Su. Robust detection of watermarks in large language models under human edits. *arXiv preprint arXiv:2411.13868*, 2024b.

X. Li, F. Ruan, H. Wang, Q. Long, and W. J. Su. A statistical framework of watermarks for large language models: Pivot, detection efficiency and optimal rules. *The Annals of Statistics*, 53(1):322–351, 2025a.

X. Li, G. Wen, W. He, J. Wu, Q. Long, and W. J. Su. Optimal estimation of watermark proportions in hybrid AI–human texts. *arXiv preprint arXiv:2506.22343*, 2025b.

W. Liang, Z. Izzo, Y. Zhang, H. Lepp, H. Cao, X. Zhao, L. Chen, H. Ye, S. Liu, Z. Huang, et al. Monitoring AI-modified content at scale: A case study on the impact of ChatGPT on AI conference peer reviews. In *International Conference on Machine Learning*, 2024.

E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*, 2023.

J. Neyman. Smooth test for goodness of fit. *Scandinavian Actuarial Journal*, 1937(3-4):149–199, 1937.

References V

L. Pan, A. Liu, Z. He, Z. Gao, X. Zhao, Y. Lu, B. Zhou, S. Liu, X. Hu, L. Wen, et al. MarkLLM: An Open-Source Toolkit for LLM Watermarking. In *Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 61–71, 2024.

K. Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

J. Piet, C. Sitawarin, V. Fang, N. Mu, and D. Wagner. Mark my words: Analyzing and evaluating language model watermarks. *arXiv preprint arXiv:2312.00273*, 2023.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21 (1):5485–5551, 2020.

N. V. Smirnov. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Math. Univ. Moscou*, 2(2):3–14, 1939.

G. S. Watson. Goodness-of-fit tests on a circle. *Biometrika*, 48(1/2):109–114, 1961.

Y. Wu, Z. Hu, H. Zhang, and H. Huang. DiPmark: A stealthy, efficient and resilient watermark for large language models. *arXiv preprint arXiv:2310.07710*, 2023.

Y. Xie, X. Li, T. Mallick, W. J. Su, and R. Zhang. Debiasing watermarks for large language models via maximal coupling. *arXiv preprint arXiv:2411.11203*, 2024.

References VI

X. Yang, W. Cheng, L. Petzold, W. Y. Wang, and H. Chen. DNA-GPT: Divergent n-gram analysis for training-free detection of GPT-generated text. *arXiv preprint arXiv:2305.17359*, 2023.

ZeroGPT. ZeroGPT: Trusted GPT-4, ChatGPT and AI detector tool by ZeroGPT. `https://www.zerogpt.com/`, 2023.

S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

X. Zhao, P. V. Ananth, L. Li, and Y.-X. Wang. Provable robust watermarking for AI-generated text. In *International Conference on Learning Representations*, 2024a. URL `https://openreview.net/forum?id=SsmT8aO45L`.

X. Zhao, L. Li, and Y.-X. Wang. Permute-and-Flip: An optimally robust and watermarkable decoder for LLMs. *arXiv preprint arXiv:2402.05864*, 2024b.

X. Zhao, C. Liao, Y.-X. Wang, and L. Li. Efficiently identifying watermarked segments in mixed-source texts. *arXiv preprint arXiv:2410.03600*, 2024c.

X. Zhao, S. Gunn, M. Christ, J. Fairoze, A. Fabrega, N. Carlini, S. Garg, S. Hong, M. Nasr, F. Tramer, S. Jha, L. Li, Y.-X. Wang, and D. Song. SoK: Watermarking for AI-Generated Content. In *IEEE Symposium on Security and Privacy*, pages 2621–2639. IEEE Computer Society, 2025.